

# **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:

P3

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
4.15 Proxy – instalacja i konfiguracja . . . . .	55
4.16 SMTP – instalacja i konfiguracja . . . . .	56
4.17 POP-IMAP – instalacja i konfiguracja . . . . .	58

<b>5 Testy działania wdrożonych usług</b>	<b>62</b>
5.1 DNS . . . . .	62
5.2 DHCP . . . . .	63
5.3 Raid 5 . . . . .	64
5.4 Samba . . . . .	64
5.5 HTTP . . . . .	65
5.6 PHP . . . . .	66
5.7 MySQL . . . . .	66
5.8 phpMyAdmin . . . . .	67
5.9 UserDir – serwer http . . . . .	68
5.10 WordPress . . . . .	68
5.11 Proxy . . . . .	69
5.12 Poczta . . . . .	71
<b>6 Kod skryptu BASH, oraz tablica crontab</b>	<b>71</b>
<b>7 Wnioski</b>	<b>71</b>
<b>8 Literatura</b>	<b>72</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	Proxy – instalacja . . . . .	55
81	Proxy – konfiguracja część pierwsza . . . . .	55
82	Proxy – konfiguracja część druga . . . . .	56
83	Proxy – konfiguracja część druga . . . . .	56
84	Konfiguracja /etc/postfix/main.cf – część pierwsza . . . . .	57
85	Konfiguracja /etc/postfix/main.cf – część druga . . . . .	57
86	Konfiguracja /etc/dovecot/dovecot.conf . . . . .	58
87	Konfiguracja /etc/dovecot/conf.d/10-auth.conf – część pierwsza . . . . .	58
88	Konfiguracja /etc/dovecot/conf.d/10-auth.conf – część druga . . . . .	59
89	Konfiguracja /etc/dovecot/conf.d/10-mail.conf . . . . .	59
90	Konfiguracja /etc/dovecot/conf.d/10-master.conf . . . . .	60
91	Konfiguracja /etc/dovecot/conf.d/10-ssl.conf . . . . .	60
92	Restart usługi Dovecot i dodaSnie do zapory sieciowej . . . . .	61
93	Test zmiennej środowiskowej MAIL . . . . .	61
94	mailx – przykład użycia . . . . .	62
95	Test DNS . . . . .	62
96	Instalacja DHCP . . . . .	63
97	Test automatycznego montowania . . . . .	64
98	Samba – próba podłączenia się do udziału na serwerze . . . . .	64

99	Samba – wynik poprzedniego kroku . . . . .	65
100	Test działania serwera WWW . . . . .	65
101	Strona html + PHP . . . . .	66
102	Test usługi mariadb (MySQL) . . . . .	66
103	Test usługi phpMyAdmin – część pierwsza . . . . .	67
104	Test usługi phpMyAdmin – część druga . . . . .	67
105	MySQL – instalacja część druga . . . . .	68
106	Dashboard WordPress'a . . . . .	68
107	Proxy – ustawienie w FireFox . . . . .	69
108	Proxy – dostęp do strony ostrzelenie . . . . .	69
109	Proxy – wynik strony po zignorowaniu ostrzelenia . . . . .	70
110	Proxy – monitoring ruchu sieciowego z serwera . . . . .	70
111	Wysłanie maila na serwerze . . . . .	71

# 1 Założenia projektowe – wymagania

- a) Systemy operacyjne: Fedora Server 39 lub inny serwer z rodziną 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**

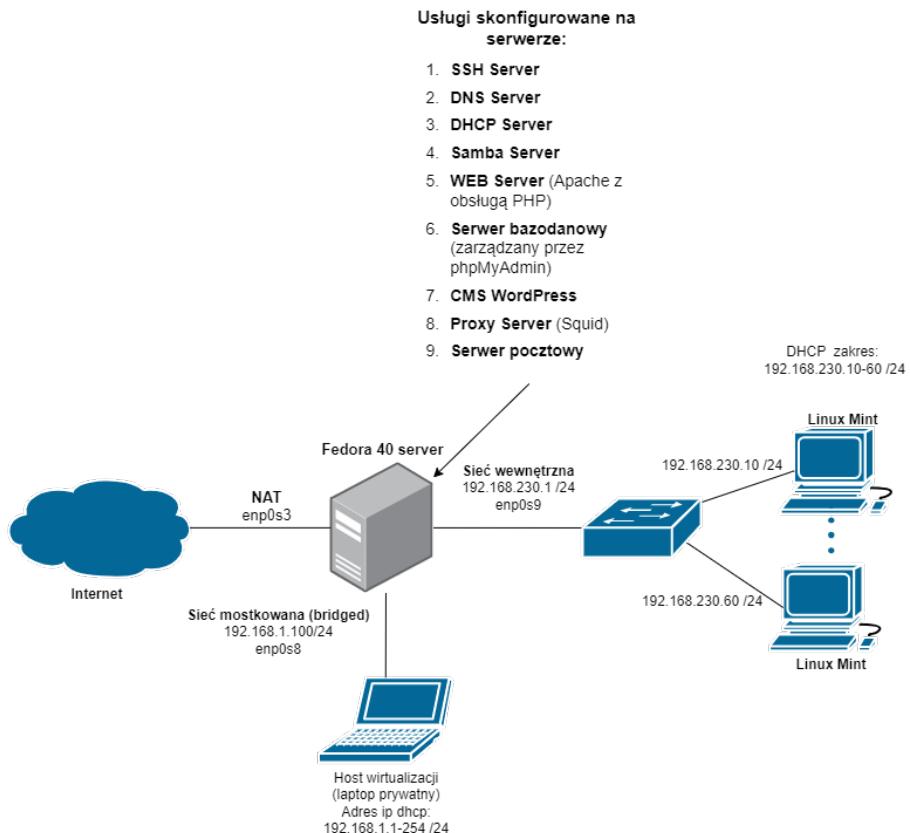
### **2.13.1 Skrypty Bash**

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

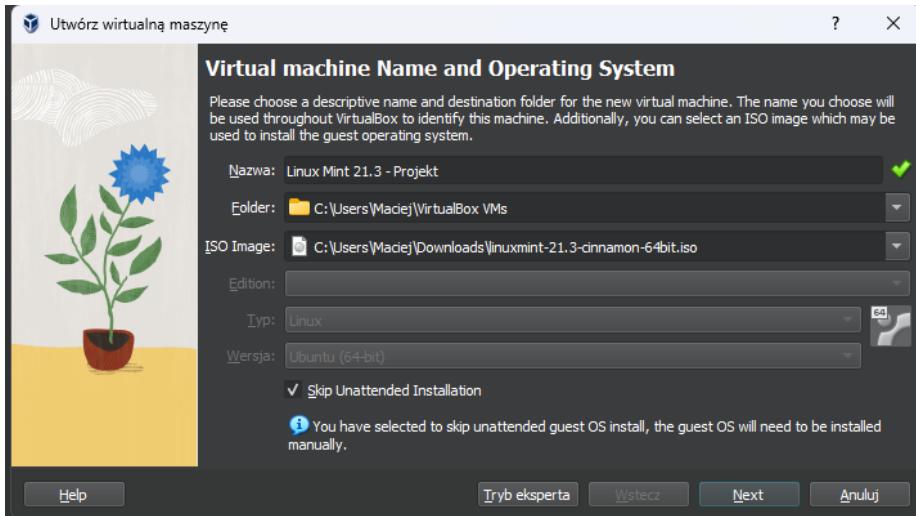


Rysunek 1: Schemat logiczny sieci

## 4 Procedury instalacyjne poszczególnych usług

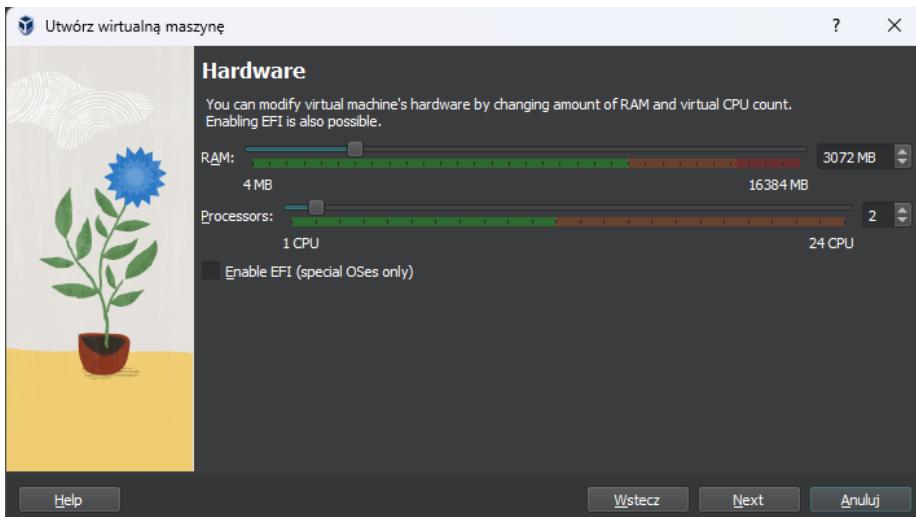
### 4.1 Instalacja systemu klienta – Linux Mint

#### 4.1.1 Proces instalacji



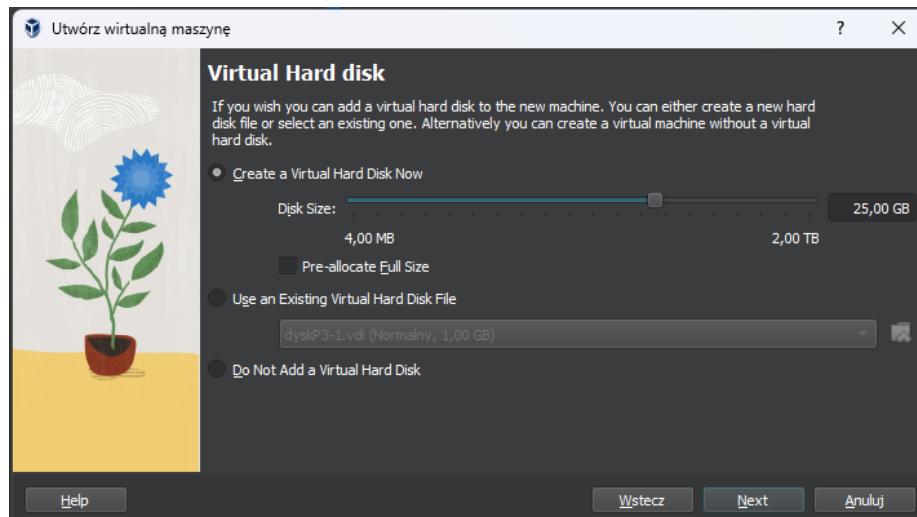
Rysunek 2: Tworzenie nowej maszyny wirtualnej. Ustawienia nazwy, lokalizacji dysku oraz wybór pliku ISO systemu operacyjnego.

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.



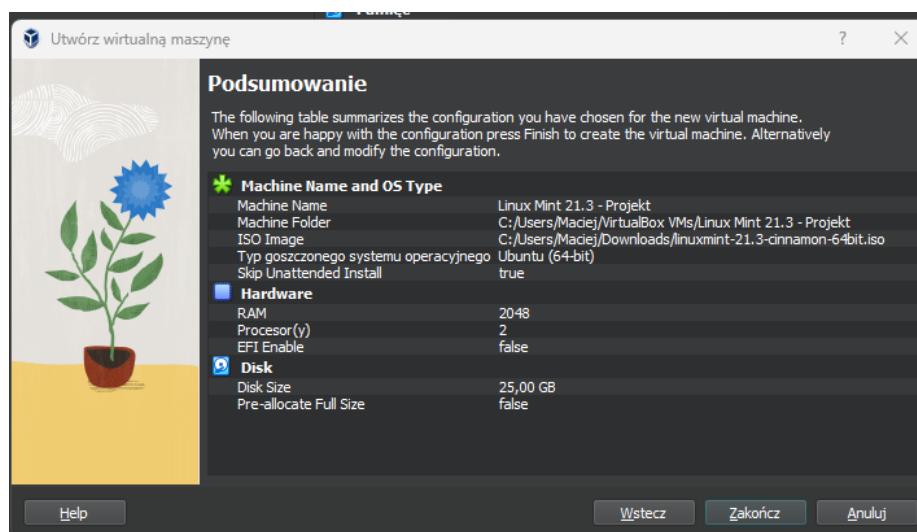
Rysunek 3: Przydzielanie zasobów maszynie wirtualnej, takich jak pamięć RAM i procesor.

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



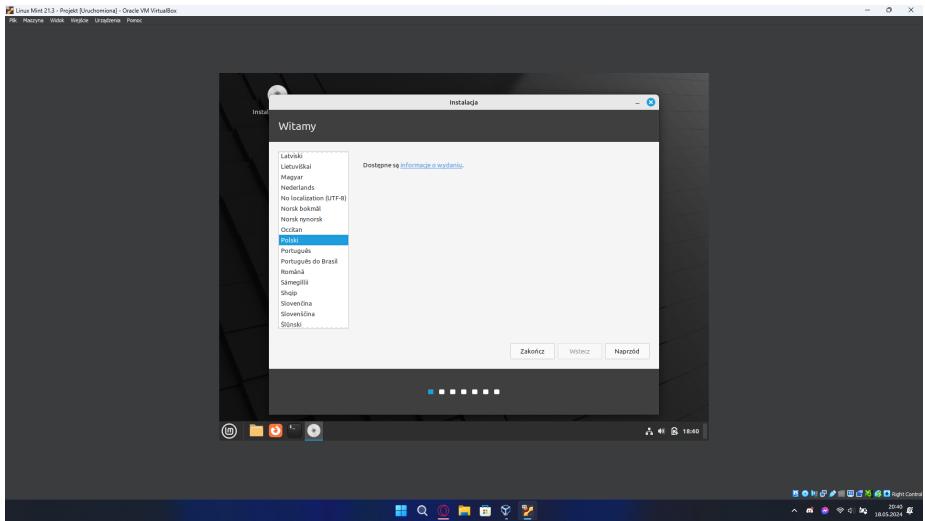
Rysunek 4: Określenie rozmiaru dysku wirtualnego.

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



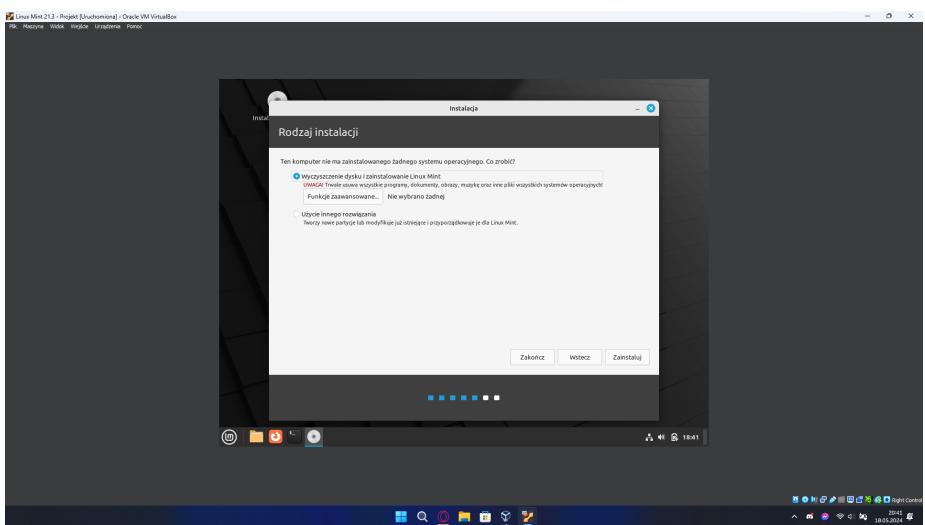
Rysunek 5: Podsumowanie konfiguracji maszyny wirtualnej przed rozpoczęciem instalacji systemu.

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



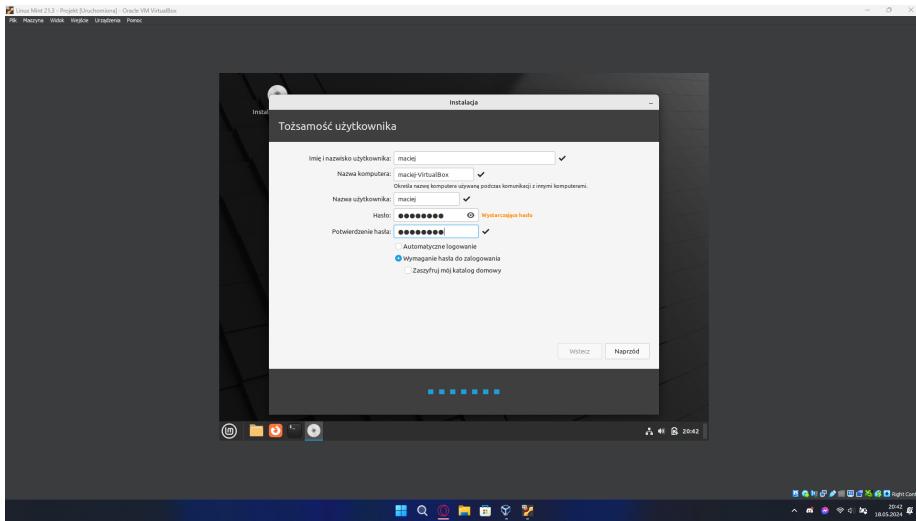
Rysunek 6: Rozpoczęcie instalacji Linux Mint – wybór języka instalacji.

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



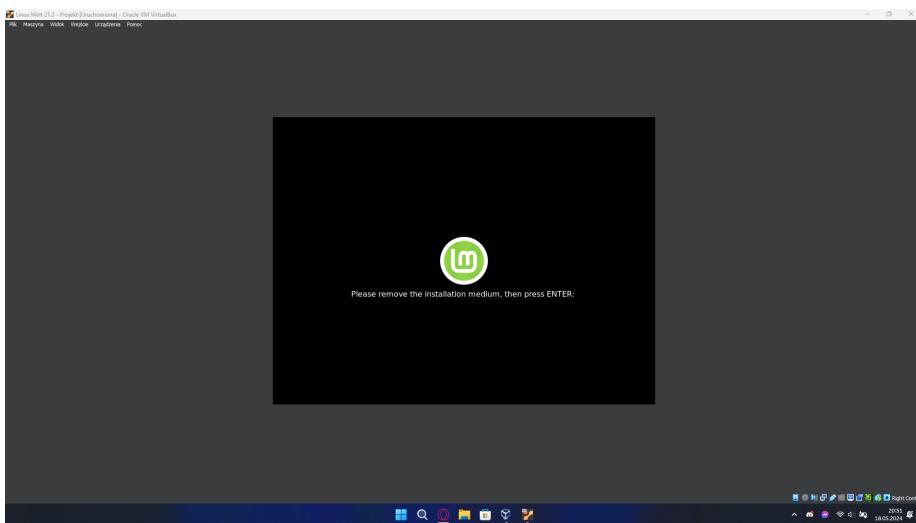
Rysunek 7: Wybór trybu instalacji na dysku twardym.

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.

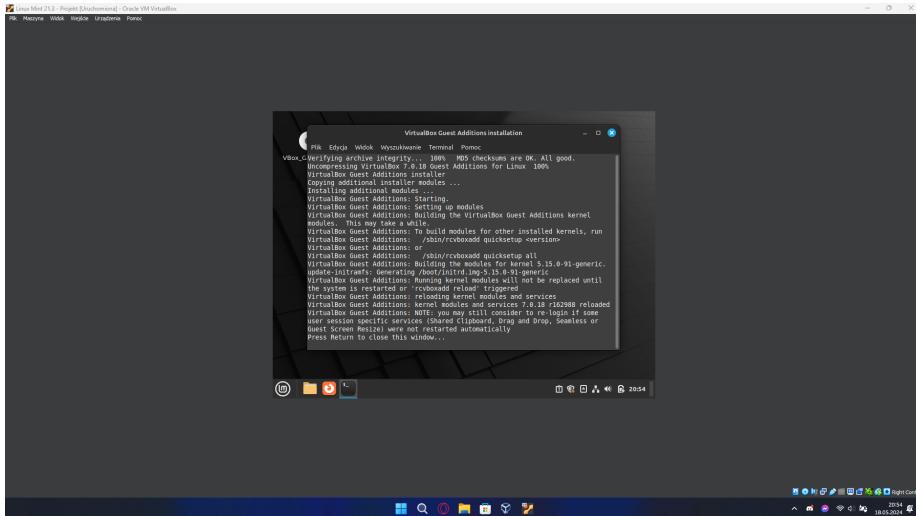
Kolejnym krokiem jest utworzenie konta użytkownika, wprowadzenie nazwy użytkownika, hasła oraz nazwy komputera.



Rysunek 9: Zakończenie instalacji systemu Linux Mint.

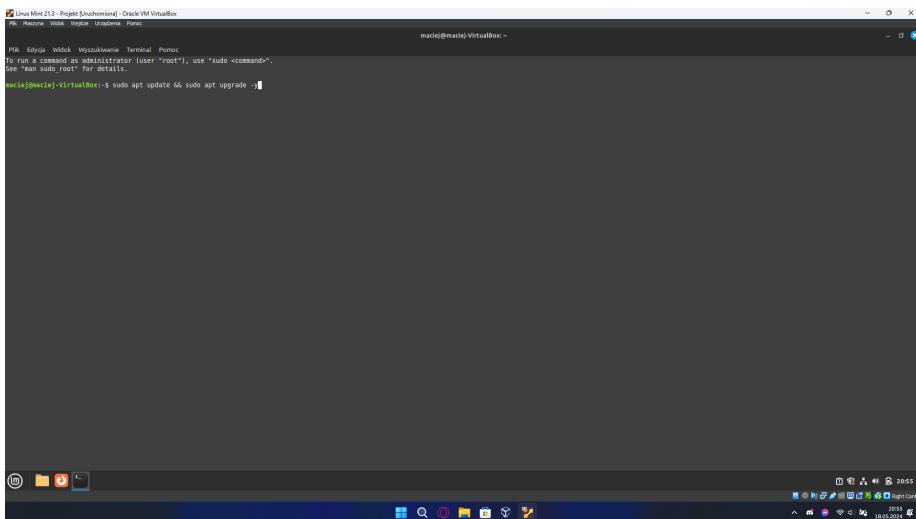
Wyświetlony zostaje monit z prośbą o usunięcie nośnika instalacyjnego. Po zakończeniu instalacji system wyświetla ekran informujący o pomyślnym zakończeniu procesu.

#### 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.

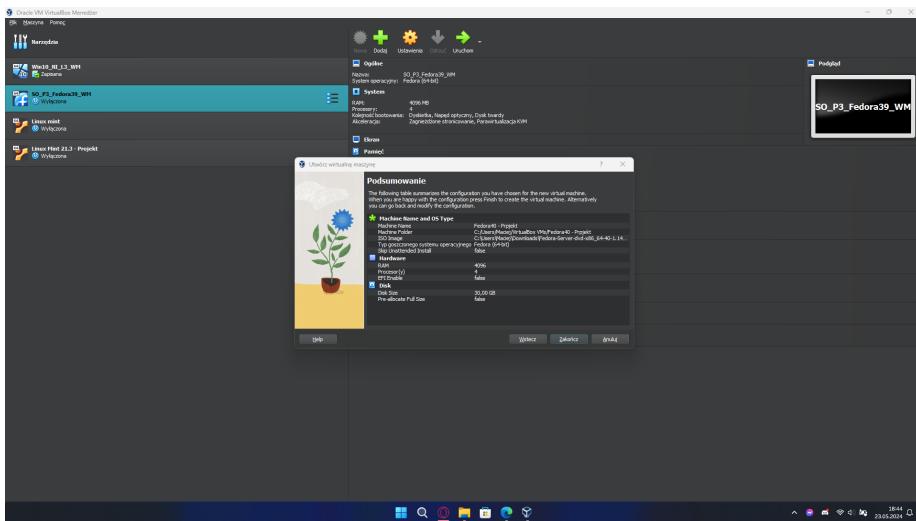


Rysunek 11: Aktualizacja pakietów systemowych.

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

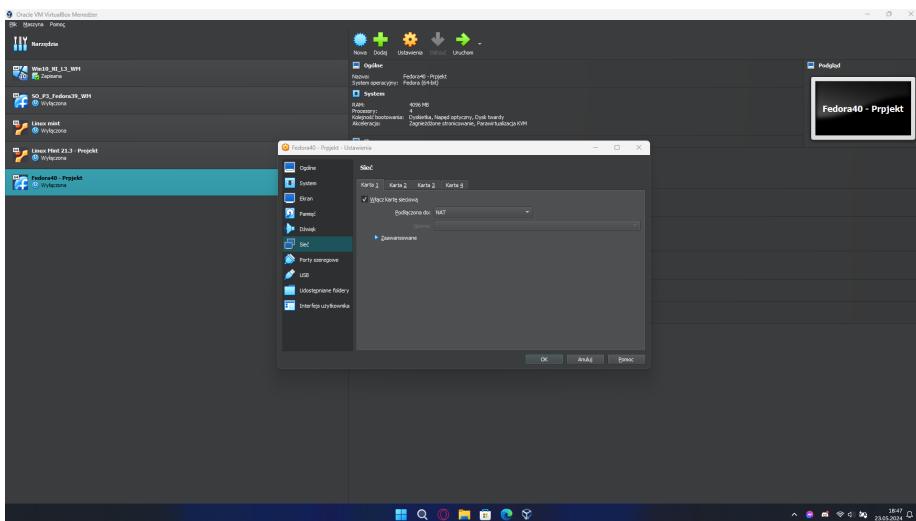
## 4.2 Instalacja serwera – Fedora 40

#### 4.2.1 Proces instalacji



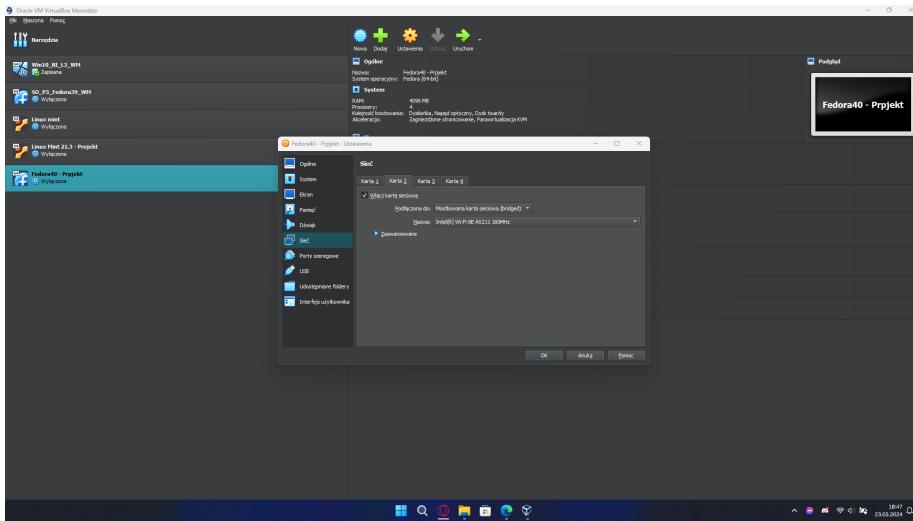
Rysunek 12: Analogicznie jak w przypadku instalacji Linux Mint – wymagane jest ustawienie nazwy maszyny wirtualnej, przydzielenie jej zasobów, ustalenie rozmiaru dysku. Powyższe zdjęcie ukazuje ekran z podsumowaniem wybranych opcji

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



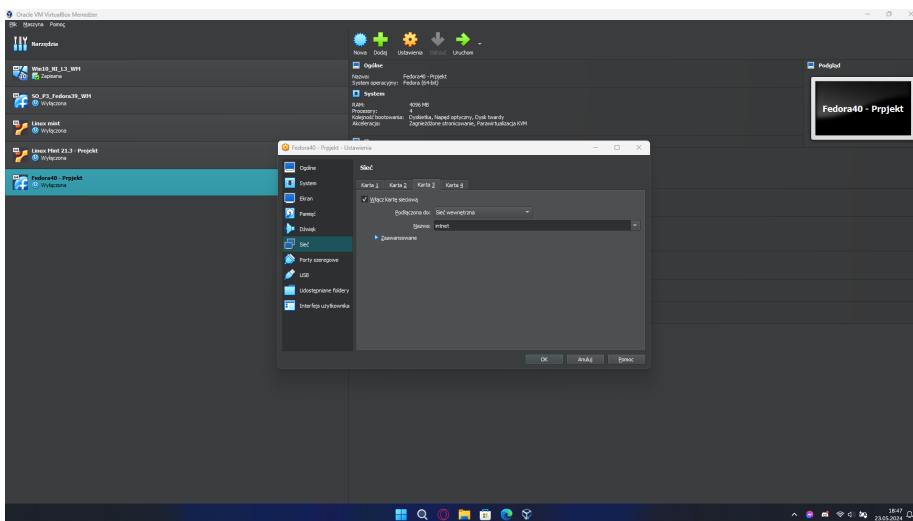
Rysunek 13: Dodanie pierwszej karty sieciowej – NAT

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.



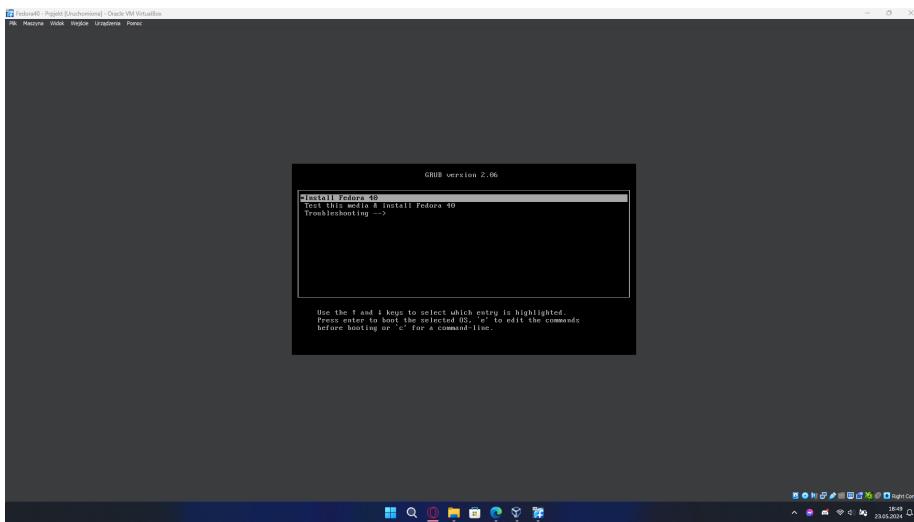
Rysunek 14: Dodanie drugiej karty sieciowej – sieć mostkowana (bridged)

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.



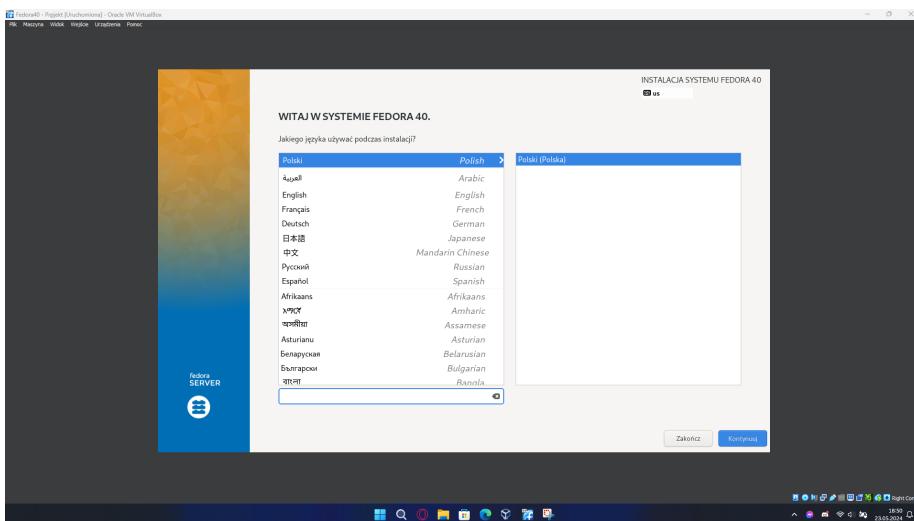
Rysunek 15: Dodanie trzeciej karty sieciowej – sieć wewnętrzna

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.



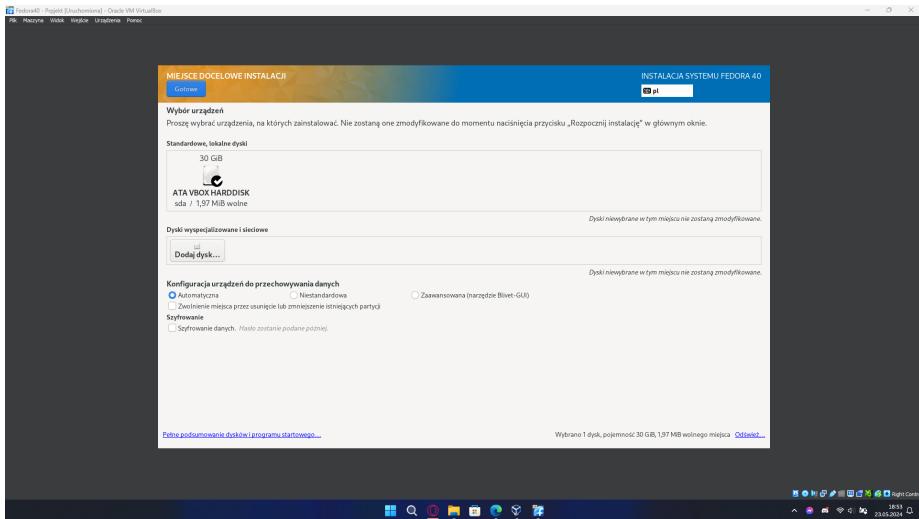
Rysunek 16: Uruchomienie instalatora Fedory.

W następnym kroku wybiera się język instalatora oraz układ klawiatury.



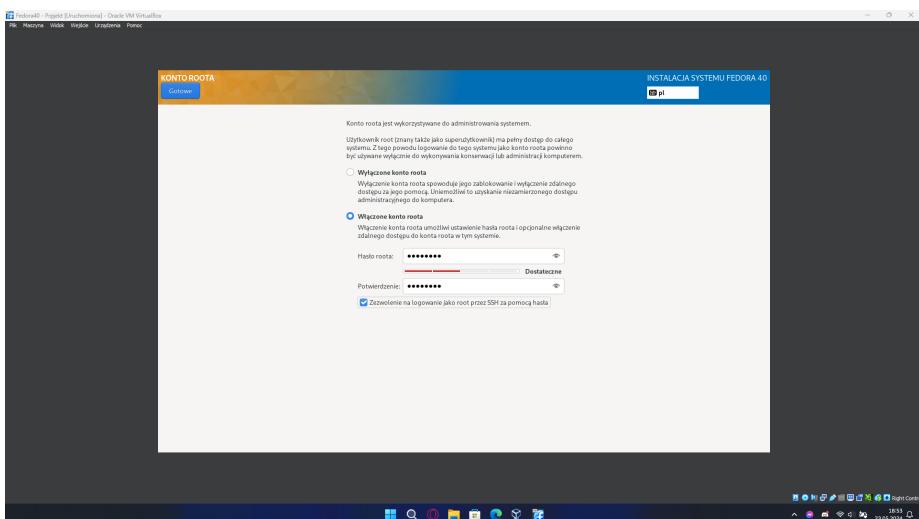
Rysunek 17: Rozpoczęcie instalacji Fedora 40 – wybór języka instalacji.

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).



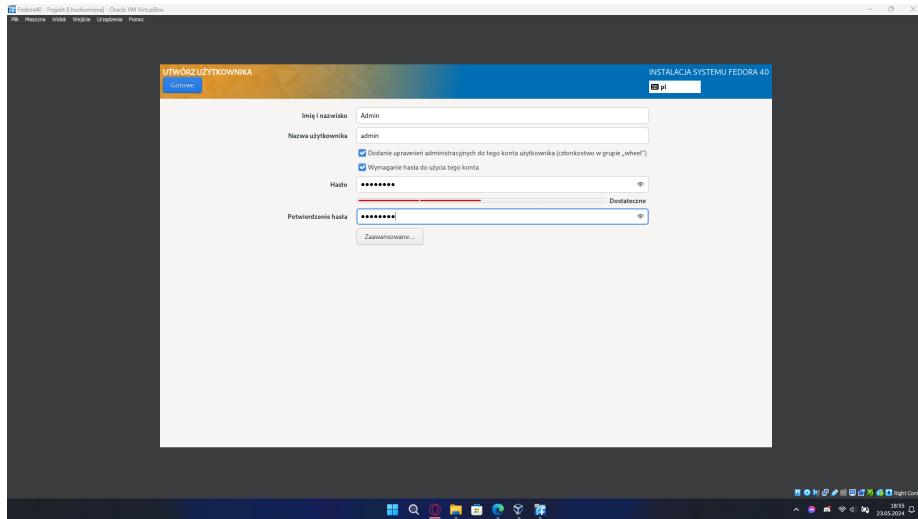
Rysunek 18: Wybór dysku na którym zostanie zainstalowany serwer

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.



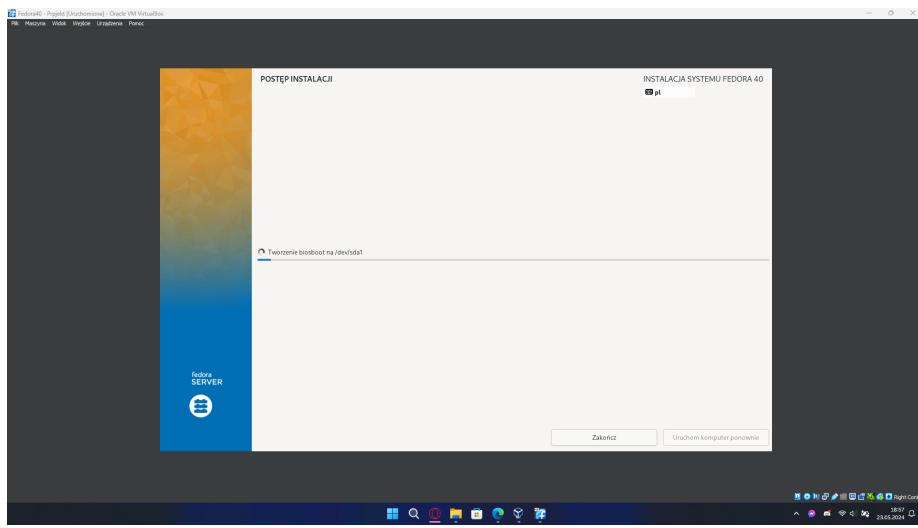
Rysunek 19: Ustawienie konta root – włączenie konta, ustawienie hasła i zezwolenie na połączenie ssh jako root

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).



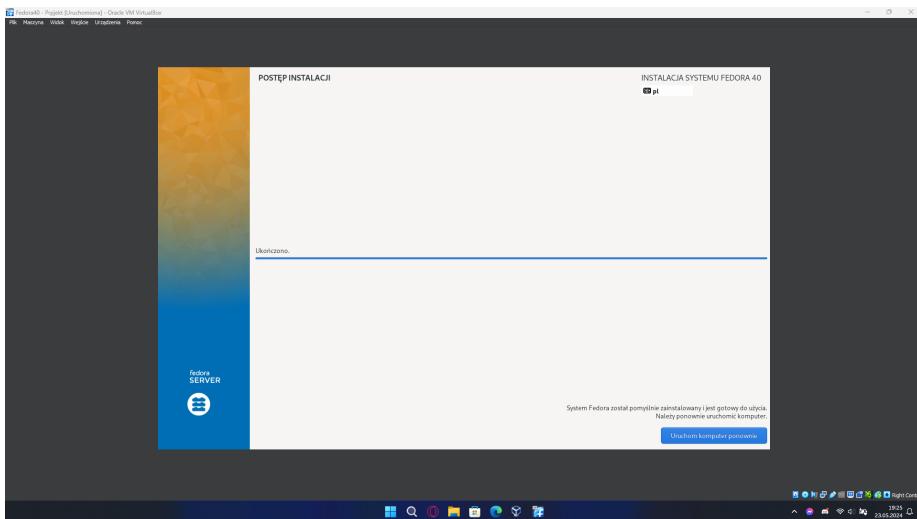
Rysunek 20: Stworzenie użytkownika – admin

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



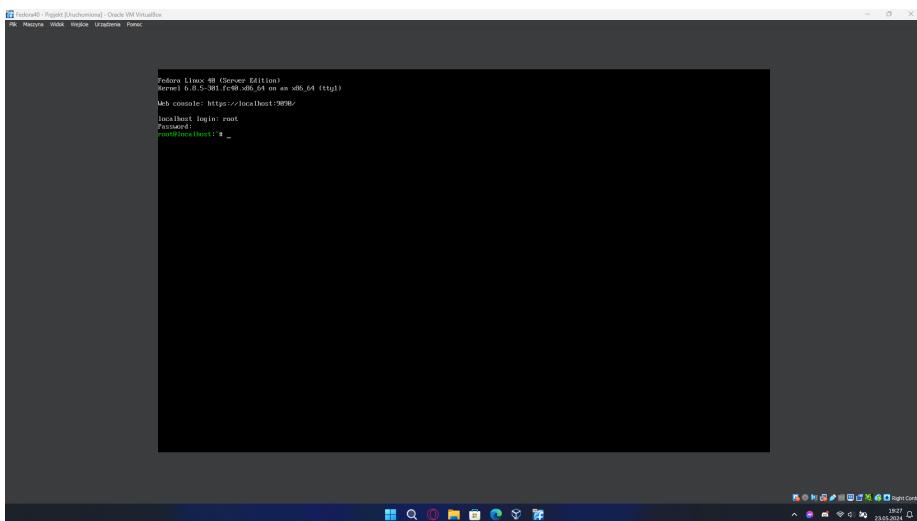
Rysunek 21: Ekran postępującej instalacji

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

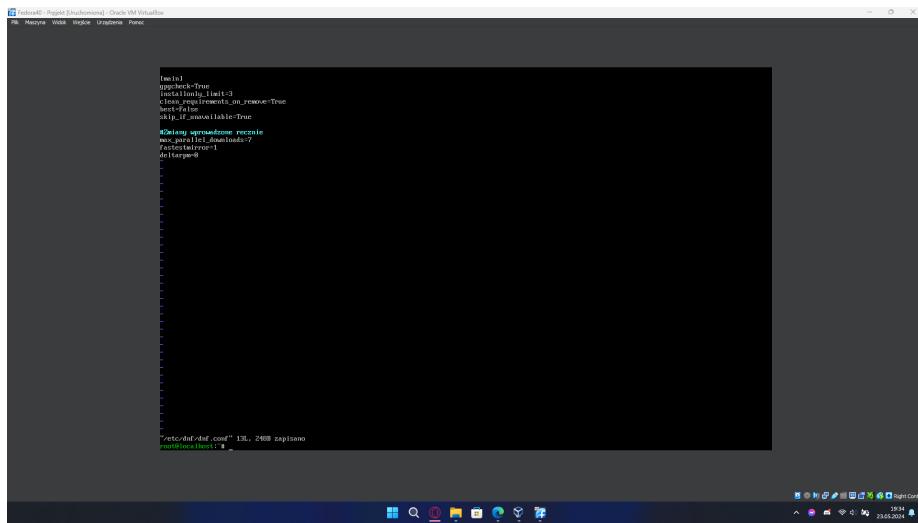
#### 4.2.2 Wstępna konfiguracja

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

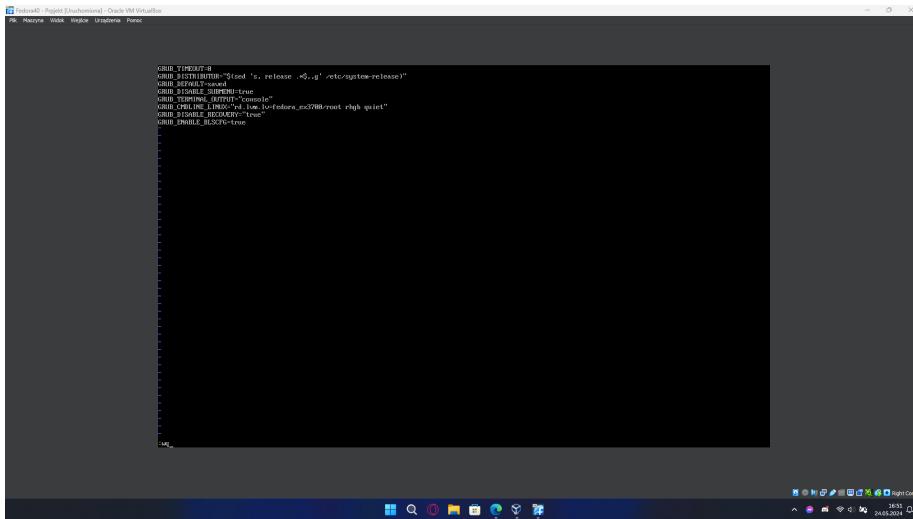
Teraz po skonfigurowaniu menadżera pakietów można wykonać aktualizację pakietów.

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)

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

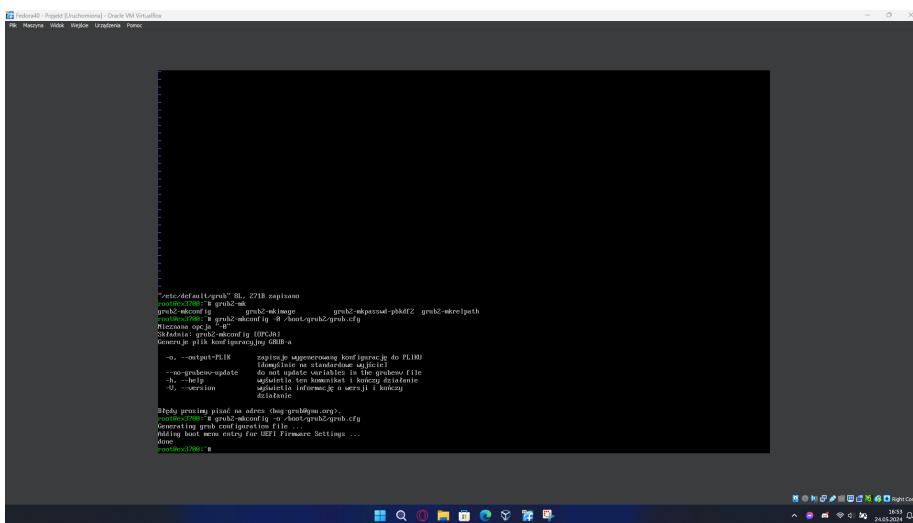
W kolejnym kroku zmieniłem GRUB\_TIMEOUT=5 na GRUB\_TIMEOUT=0  
Co można zauważyc poniższym zdjeciu.



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

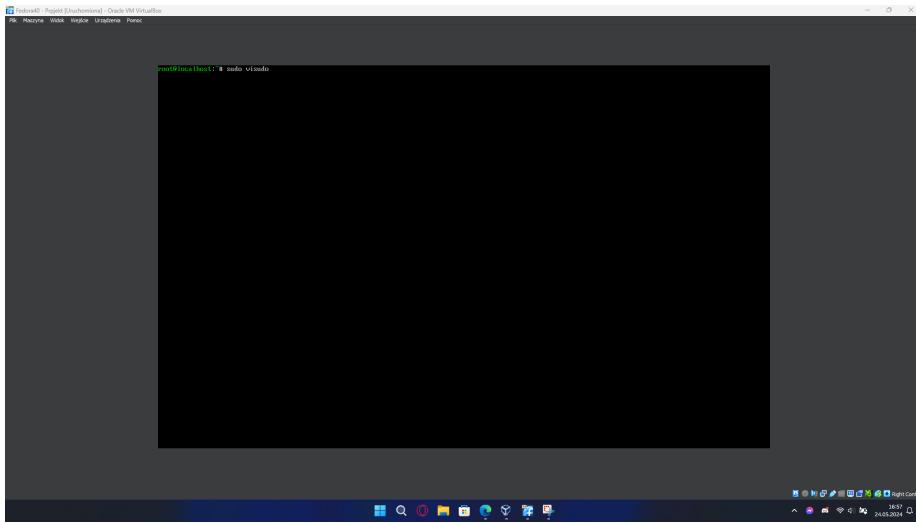
Aby zatwierdzić zmiany należy użyć komendy:

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



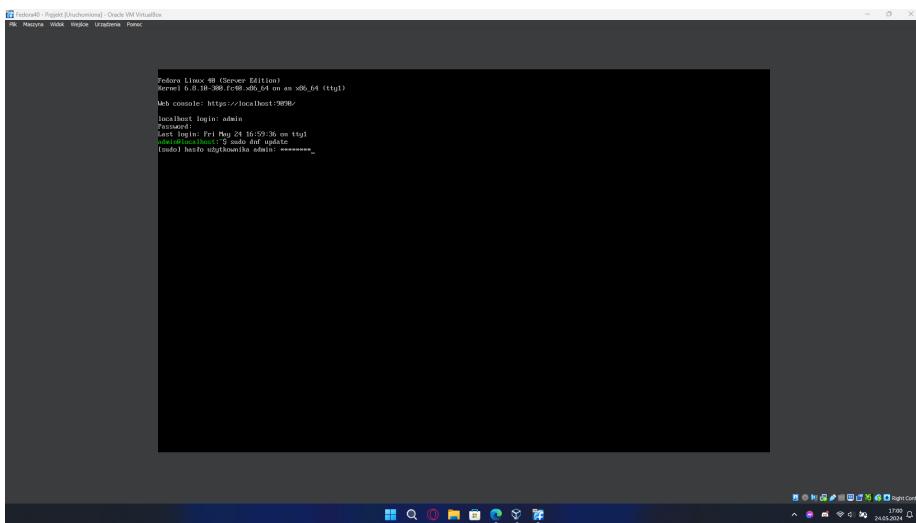
Rysunek 28: Zastosowanie zmian po edycji grub

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



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

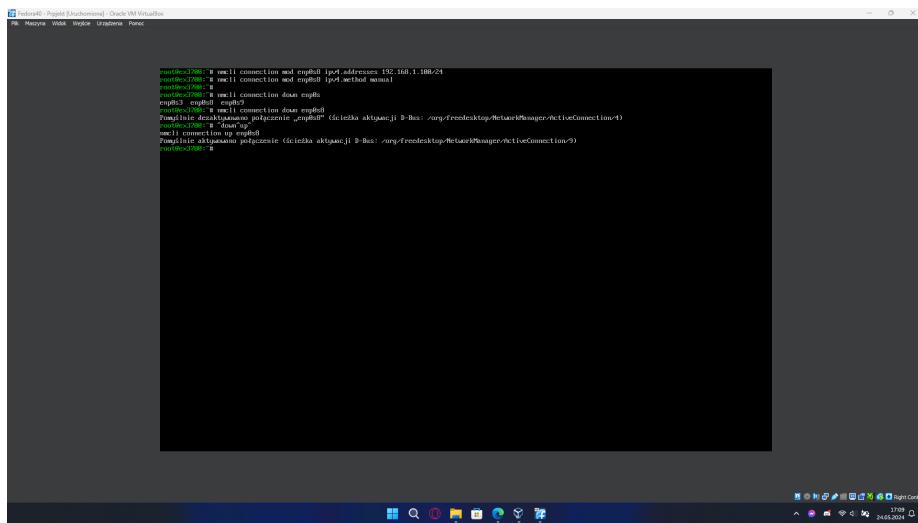
Efekt powyższego kroku:



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

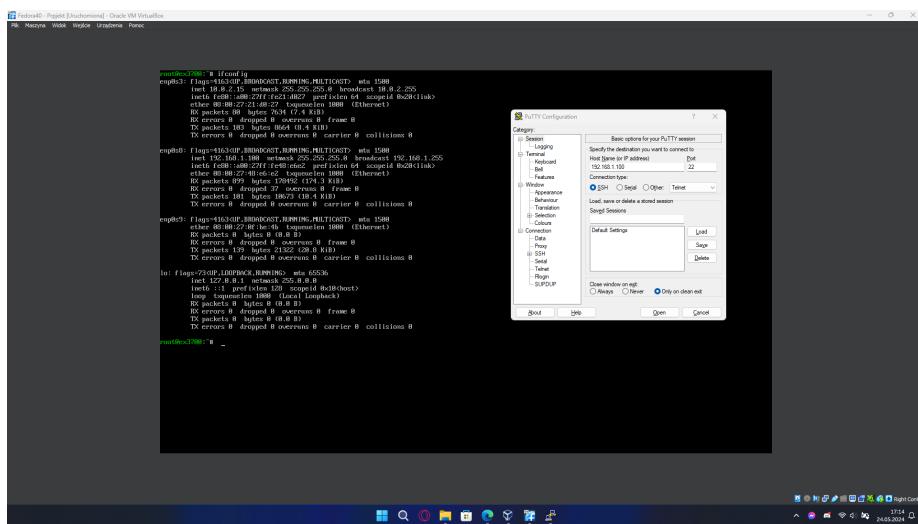
## 4.3 Konfiguracja SSH

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)



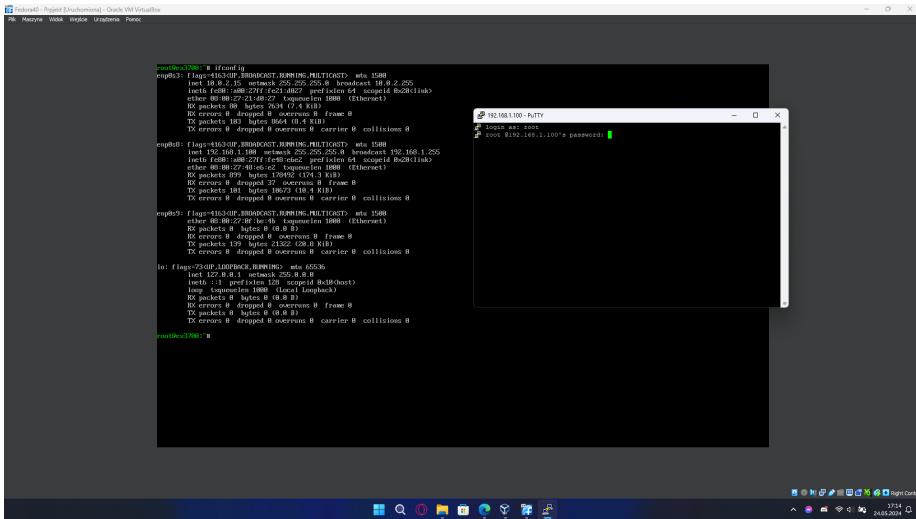
Rysunek 31: konfiguracja karty sieciowej

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



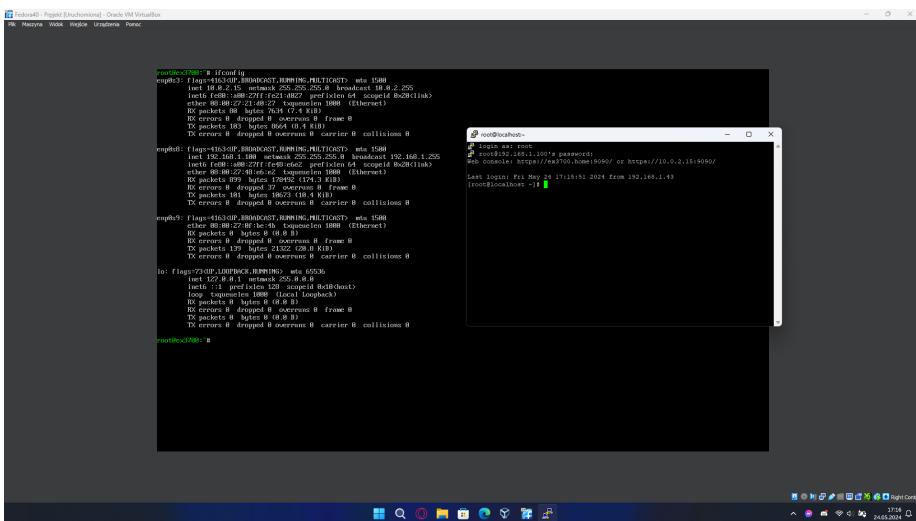
Rysunek 32: Konfiguracja aplikacji PuTTY

Próba zalogowania na konto root'a:



Rysunek 33: Podlaczenie poprzez PuTTY na konto root'a

Wynik powyższego kroku:

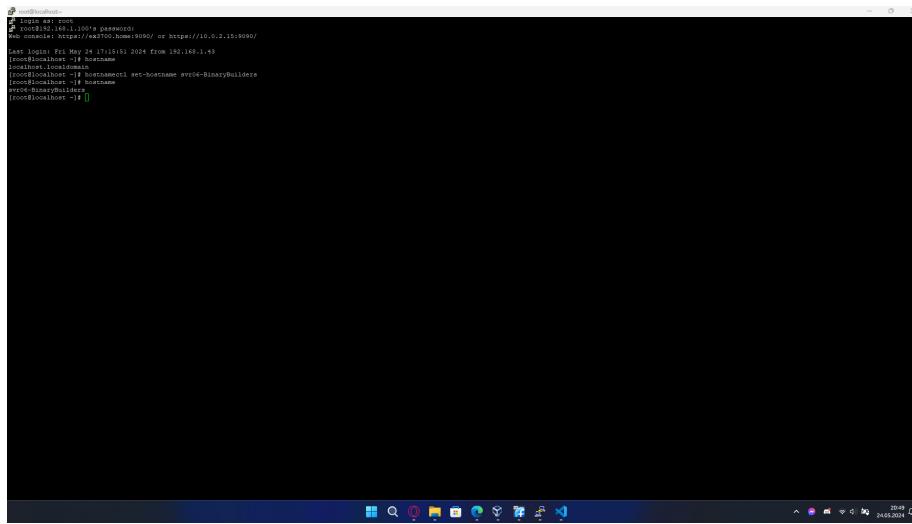


Rysunek 34: Wynik połączenia poprzez PuTTY

## 4.4 Nazwa serwera – hostname

Aby zmienić nazwę serwera (hostname) można użyć komendy:

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



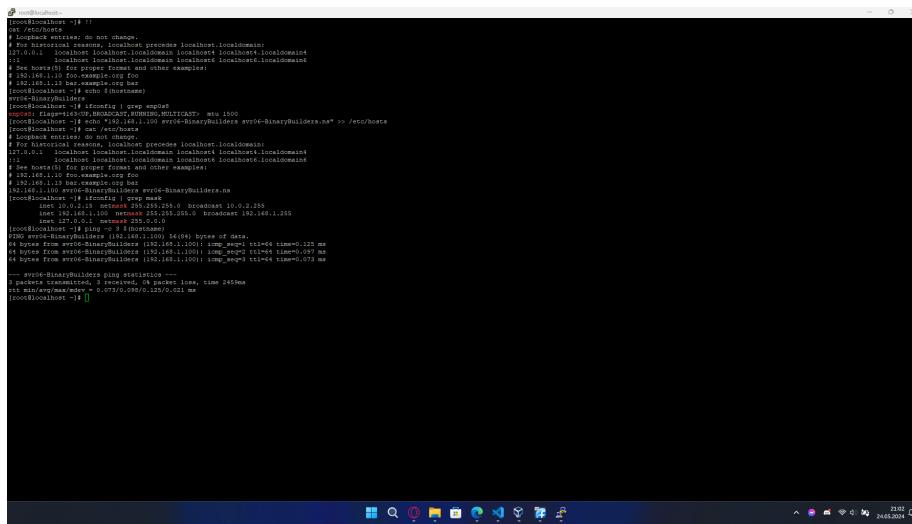
```
[root@svr04 ~]# hostnamectl set-hostname svr04-BinaryBuilders
[root@svr04 ~]#
```

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
127.0.0.1 localhost localhost.localdomain localhost.localdomain4
::1 localhost localhost.localdomain localhost localhost.localdomain6
192.168.1.11 bar.example.org bar
192.168.1.12 www.example.org www
svr04-BinaryBuilders
[root@svr04 ~]# echo "192.168.1.100 svr06-BinaryBuilders svr06-BinaryBuilders.ns" >> /etc/hosts
[root@svr04 ~]# cat /etc/hosts
127.0.0.1 localhost localhost.localdomain localhost localhost.localdomain4
::1 localhost localhost.localdomain localhost localhost.localdomain6
192.168.1.11 bar.example.org bar
192.168.1.12 www.example.org www
192.168.1.100 svr06-BinaryBuilders svr06-BinaryBuilders.ns
[root@svr04 ~]# ifconfig | grep eth0
eth0      Link encap:Ethernet HWaddr 00:0C:29:1A:0D:01
          inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
              ... (redacted)
          inet 127.0.0.1 netmask 255.0.0.0
          brd 127.255.255.255
          ...
[root@svr04 ~]# ping -c 1 svr06-BinaryBuilders.ns
PING svr06-BinaryBuilders (192.168.1.100) 54(84) bytes of data:
44 bytes from svr06-BinaryBuilders (192.168.1.100): icmp_seq=1 ttl=64 time=0.125 ms
44 bytes from svr06-BinaryBuilders (192.168.1.100): icmp_seq=2 ttl=64 time=0.097 ms
44 bytes from svr06-BinaryBuilders (192.168.1.100): icmp_seq=3 ttl=64 time=0.073 ms
...
3 packets transmitted, 3 received, 0% packet loss, time 240ms
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
```

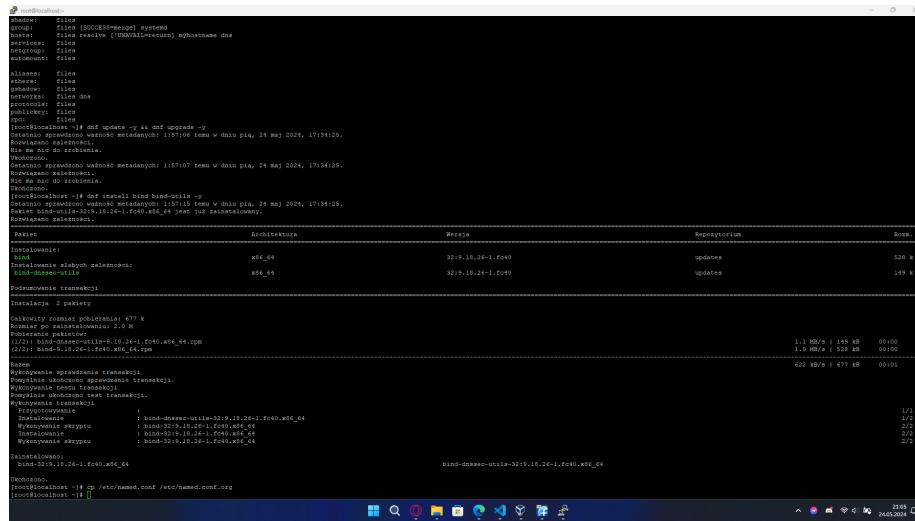
Po zainstalowaniu wymaganych pakietów należy wykonać kopię zapasową plików konfiguracyjnych. Można to zrobić komendą:

- 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";
};

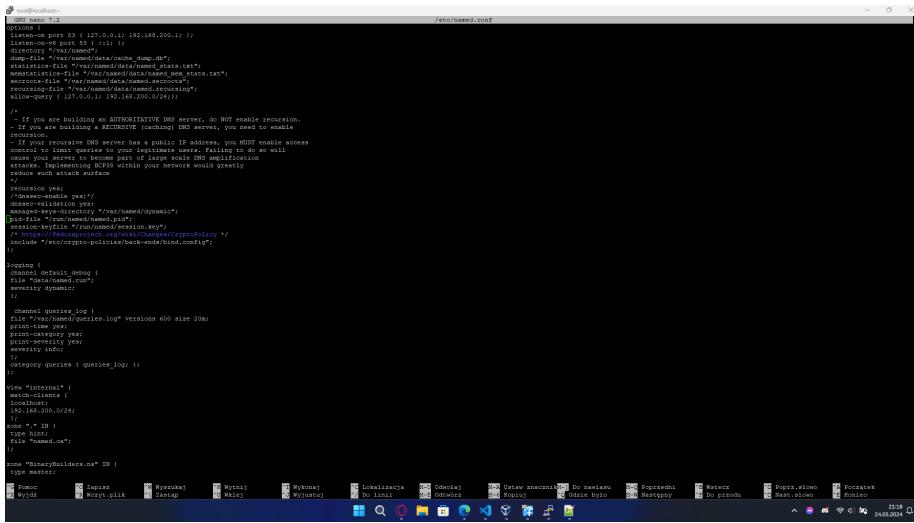
zone "BinaryBuilders.ns" IN {
    type master;
    file "BinaryBuilders.ns.lan_in";
    allow-update { none; };
};

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.

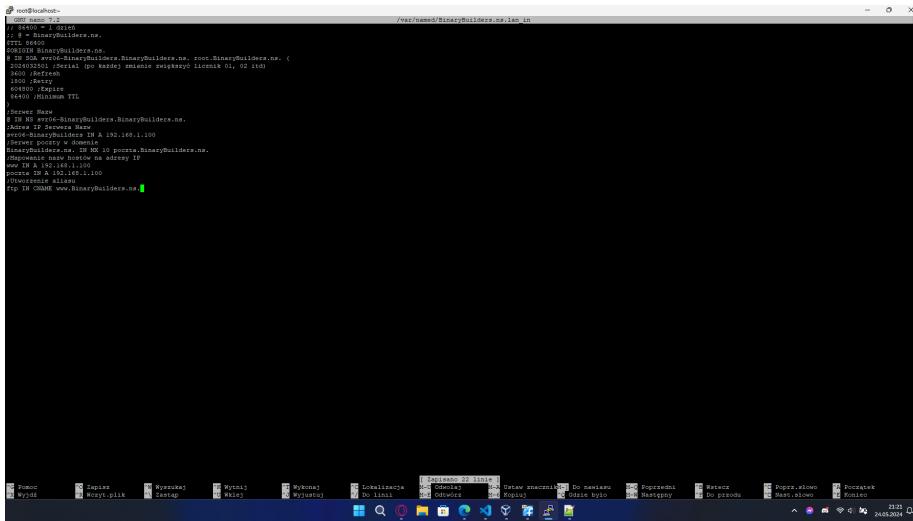
W kolejnym kroku trzeba utworzyć plik strefy podstawowej. W moim przypadku jest to plik /var/BinaryBuilders.ns.lan.in. Zawartość tego pliku:



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.

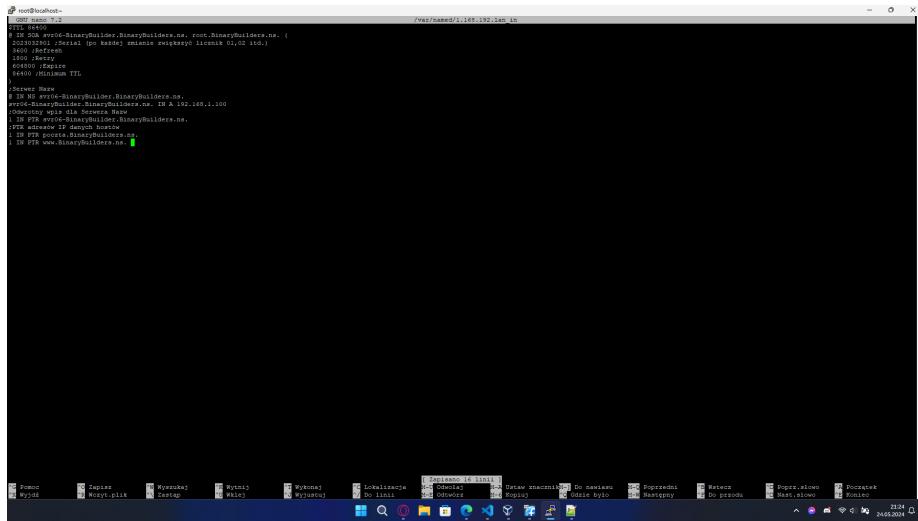
```

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

- ## • undo/reload

Przeładowuj konfigurację serwera serwera DNS

- www.dz-studium

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

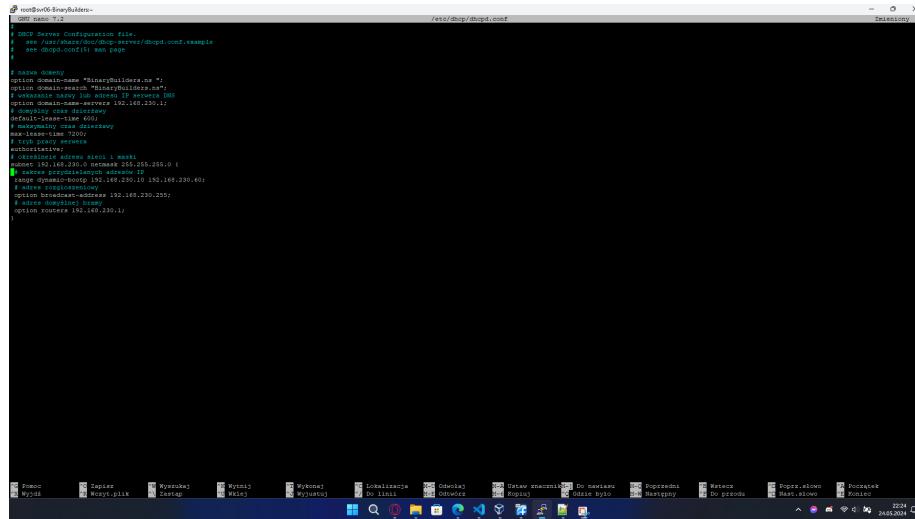
Po zainstalowaniu odpowiednich pakietów dobrze jest wykonać kopię zapasową oryginalnego pliku konfiguracji, co widać na zrucie powyżej. Można to zrobić następującą komendą:

```
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
```

Wytlumaczenie 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.

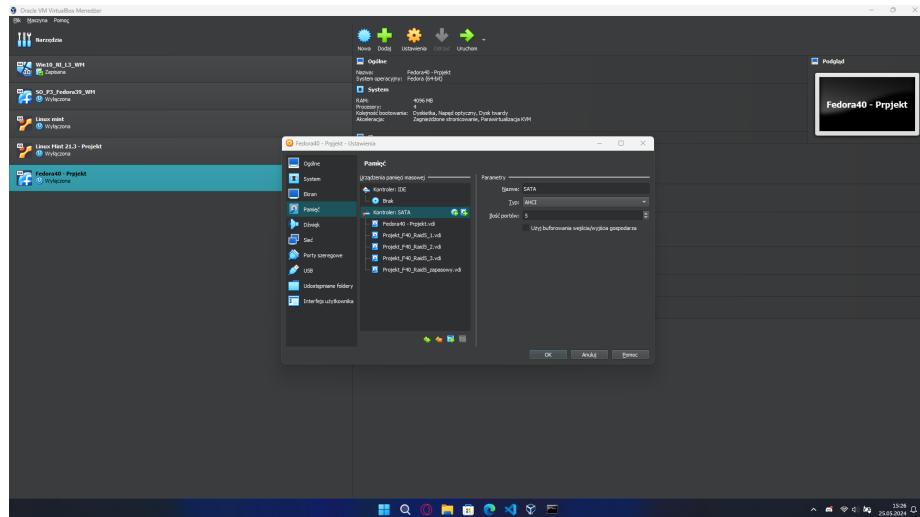
```
[root@srv06-BinaryBuilders ~]# nano /etc/dhcp/dhcpd.conf  
[root@srv06-BinaryBuilders ~]# systemctl start dhcpcd  
[root@srv06-BinaryBuilders ~]# systemctl enable dhcpcd  
[root@srv06-BinaryBuilders ~]# useradd -r -u 1000 -g 1000 -s /usr/bin/false -c 'dhcp' ventus/dhcpd.service - /usr/lib/systemd/system/dhcpd.service  
[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.

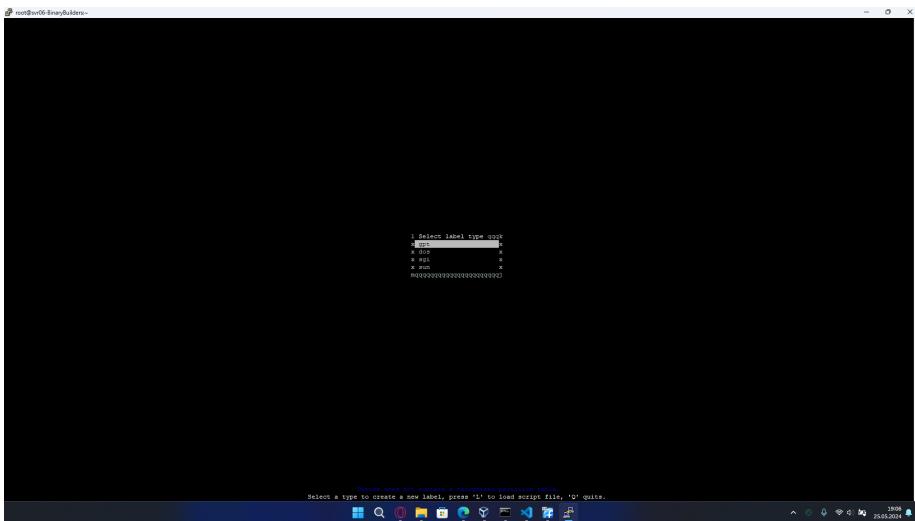
- mdadm -D /dev/md0

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.

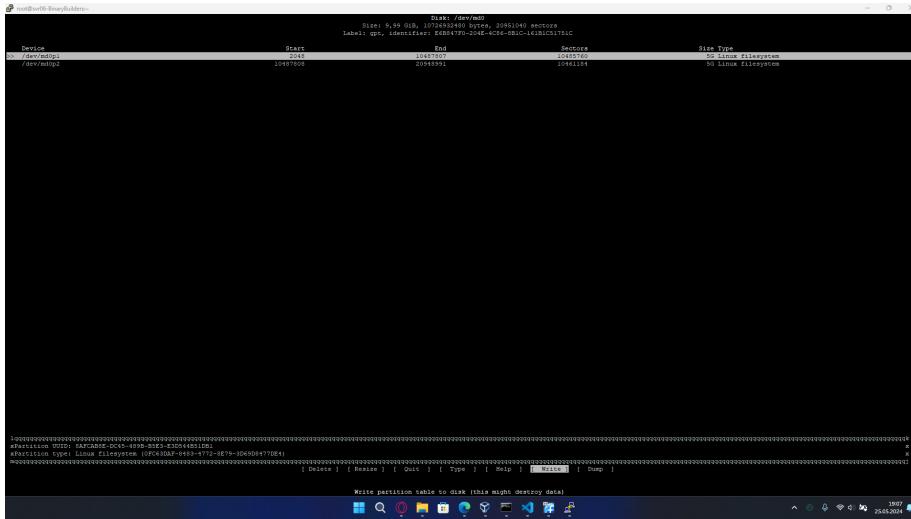
Rysunek 47: Stworzenie macierzy raid 5

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



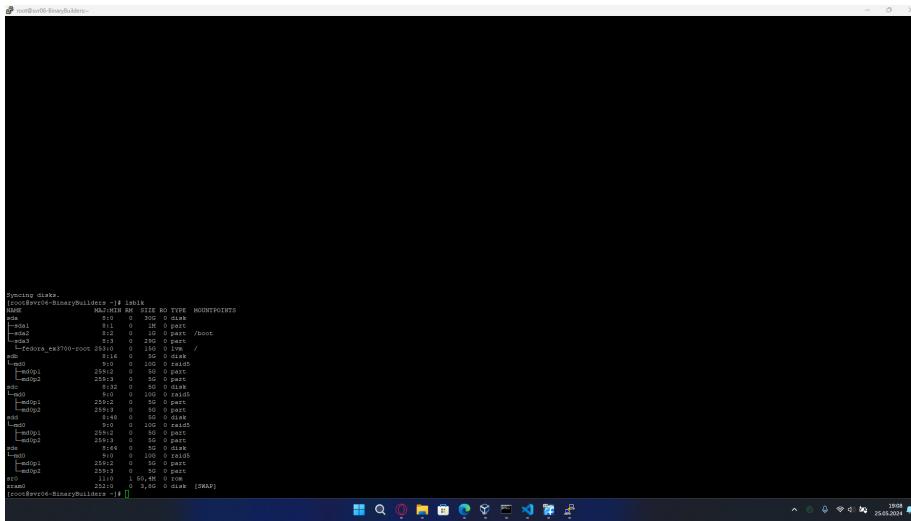
Rysunek 48: Partycjonowanie macierzy narzędziem cfdisk

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

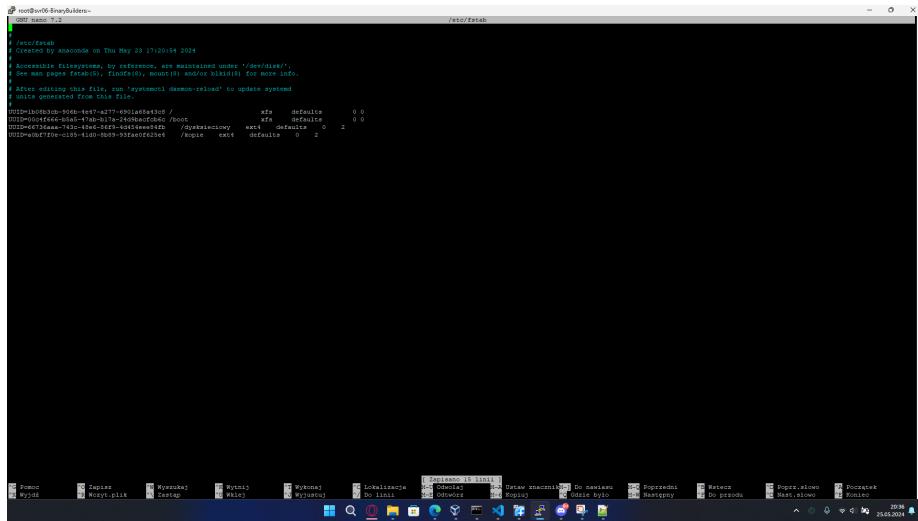
Formatowanie przed chwilą stworzonych partycji (w systemie plików ext4), stworzenie katalogów /dysksieciowy, /kopie, zamontowanie partycji do tych katalogów oraz wyświetlenie id dysków i partycji w systemie.

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-24d9bacfcfb6c /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](#).

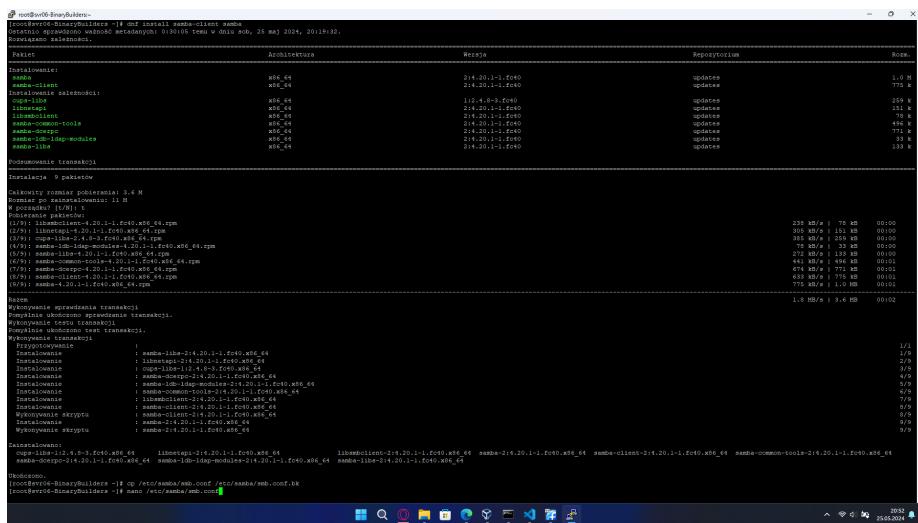
## 4.8 Samba – instalacja i konfiguracja

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
```



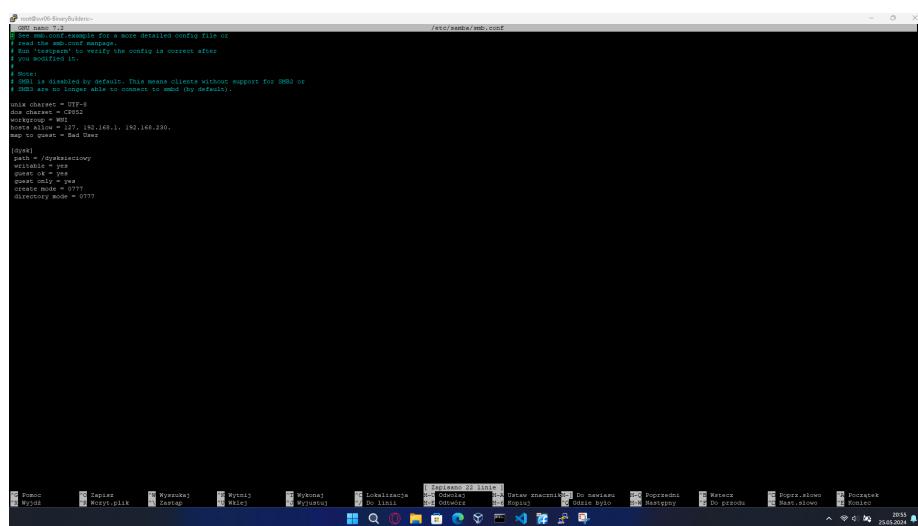
Rysunek 53: Samba – instalacja

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

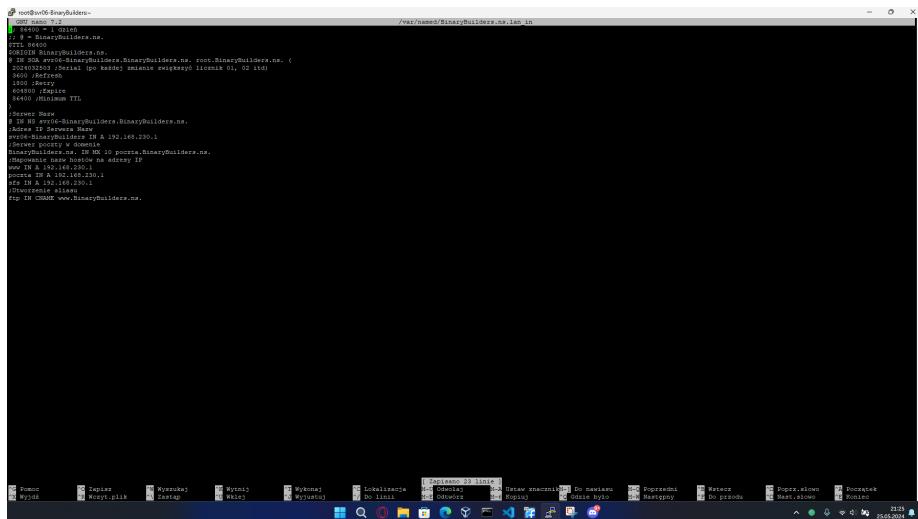
Ustawia w SELinux politykę, która pozwala Sambie na eksportowanie wszystkich udziałów z prawami do odczytu i zapisu. Dzięki temu Samba może zarządzać plikami z pełnym dostępem zgodnie z ustawieniami SELinux.

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 restarcie 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 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

```

root@w06-dvdbulden:~#
root@w06-dvdbulden:~# 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 MultiViews
6
7     # Possible values for the Options directive are "None", "All",
8     # or any combination of the keywords:
9     #   Indexes FileInfo AuthConfig Limit
10    # Note that "MultiViews" must be named *explicitly* --- "Options All"
11    # doesn't give it to you.
12
13    # The Options directive is both complicated and important. Please see
14    # http://httpd.apache.org/docs/2.4/sections/mods/section-options.html#options
15    # for more information.
16
17    Options Indexes FollowSymLinks
18
19    # AllowOverride controls what directives may be placed in .htaccess files.
20    # It can be "All", "None", or any combination of the keywords:
21    #   Options FileInfo AuthConfig Limit
22    #   AllowOverride All
23
24    # Controls who can get stuff from this server.
25    Options +Indexes All
26
27    # DirectoryIndex sets the file that Apache will serve if a directory
28    # is requested.
29
30 <Directory ~>
31     DirectoryIndex index.html
32 </Directory>
33
34 # The following lines prevent .htaccess and .htpasswd files from being
35 # viewed by Web clients.
36 <Files ".ht*">
37     Require all denied
38 </Files>
39
40 # ErrorLog: The location of the error log file.
41 # If you do not specify an ErrorLog directive within a <VirtualHost>
42 # container, then host's errors will be logged there and not here.
43 # LogFormat "%h %l %u %t \"%r\" %s %b" log
44 # LogFormat "%h %l %u %t \"%r\" %s %b" combined
45
46 <ErrorLog "/var/log/error_log">
47
48 <LogLevel debug> # Control the number of messages logged to the error log.
49 # Possible values include: debug, info, notice, warn, error, crit,
50 # alert, emerg.
51 <LogDir "logs">
52
53 <IfModule log_config_module>
54     # The following directives define some format nicknames for use with
55     # the LogFormat and CustomLog directives below.
56     LogFormat "%h %l %u %t \"%r\" %s %b" log
57     LogFormat "%h %l %u %t \"%r\" %s %b" combined
58
59 </IfModule>

```

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

```

root@w06-dvdbulden:~#
root@w06-dvdbulden:~# cat /etc/httpd/conf/httpd.conf
1 <ErrorLog "/var/log/error_log">
2
3 <LogLevel debug> # Control the number of messages logged to the error log.
4 # Possible values include: debug, info, notice, warn, error, crit,
5 # alert, emerg.
6 <LogDir "logs">
7
8 <IfModule log_config_module>
9     # The following directives define some format nicknames for use with
10    # the LogFormat and CustomLog directives below.
11    LogFormat "%h %l %u %t \"%r\" %s %b" log
12    LogFormat "%h %l %u %t \"%r\" %s %b" combined
13
14 </IfModule>

```

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

```
cd /etc/nginx/conf.d/
nginx -t
# GOST name: 7.2.2
#
# Filters allow you to process content before it is sent to the client.
# To parse .xhtml files for server-side includes (SSIs).
# (You will also need to add "includes" to the "Options" directive.)
#
AddType text/html .xhtml
AddType application/xhtml+xml .xhtml
/IMModule

# Specify a default port for all connections that don't have one.
# The port number of all connections by default is 80.
# default browser choice (HTTP:8080-1), or to allow the META tag
# attribute to override that choice, comment out this
# directive:
#
#default_charset UTF-8;

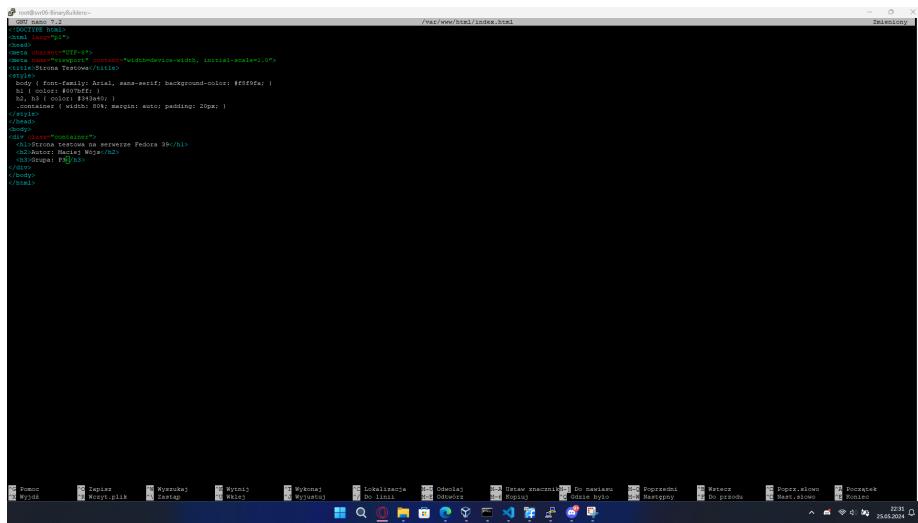
#IMModule name magic_module;
#
# The mod_magic module allows the server to use various hints from the
# contents of the file itself to determine its type. The MIMEMagicFile
# directive must be added once the hint definitions are loaded.
#
#MIMEMagicFile conf/magic
/IMModule

# Uncommentable error responses come in three flavors:
# 1) plain text 2) local references 3) external redirects
#
# Some examples:
#
#ErrorDocument 500 "The server made a boo boo."
#ErrorDocument 404 /oops/404/missing_handler.cgi
#ErrorDocument 402 http://www.example.com/subscription_info.html
#
# EnableIMAP and EnableSendfile: On systems that support it,
# these directives can greatly improve performance of IMAP and
# files. This usually improves server performance, but must
# be used with care as it can cause problems with some
# filesystems or if support for these functions is otherwise
# broken in your system.
#
# Defaults are commented. EnableIMAP On, EnableSendfile off
#
#EnableIMAP off
#EnableSendfile on
#
# Supplemental configuration
#
# Load config files in the /etc/nginx/conf.d/* directory, if any.
#IncludeOptional conf.d/*
#
server{listen Prod
        # ...
        # ...
}
```

W kolejnym 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, 12 Dec 2022 14:43:14 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 baz danych mariadb należy wydać polecenie:

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

Rysunek 65: mariadb – instalacja usług

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

character-set-server=utf8

```
root@localhost: /etc/my.cnf.d/mariadb-server.cnf
[mysqld]
# These groups are read by MySQLDB servers
# Use it for options that only the server (but not clients) should see
# See the examples of servers my.cnf files in /usr/share/mysql/
# This is read by the standalone daemon and embedded servers
[mysqld_safe]
# This is only for the mysql standalone daemon
# Settings user and group are ignored when system is used.
# Set the user name and group name to run MySQL as.
# If you do not specify a user and group,
# customize your system unit file for mysqld/mariadb according to the
# instructions in https://github.com/docker-library/mysql/blob/main/Dockerfile
[mysqld]
datadir=/var/lib/mysql
socket=/var/run/mysqld/mysqld.sock
log_error=/var/log/mariadb/mariadb.log
pid_file=/var/run/mysqld/mysqld.pid
character-set-server=utf8
collation-server=utf8_general_ci

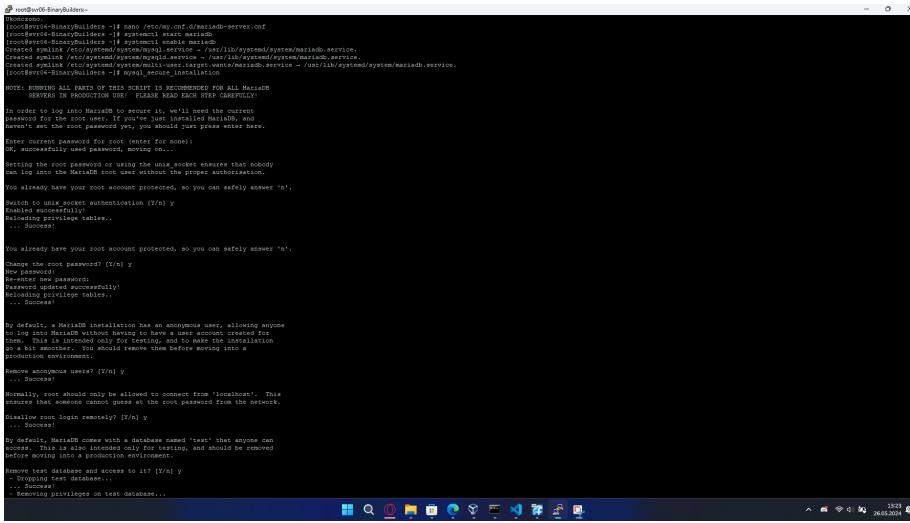
# Oracle-related settings
[galera]
# Galera specific settings
server_id=1
server_name=mariadb
server_cluster_address=127.0.0.1:4567
default_storage_engine=InnoDB
innodb_autoinc_lock_mode=2
# Allow servers to accept connections on all interfaces.
bind-address=0.0.0.0

# Optional setting
[mysqld_rpl]
rpl_recovery_log=1
rpl_max_tx_size=1000000000
rpl_max_tx_commit=1

# This is only for embedded servers
[embedded]
# This group is only read by MySQL servers, not by MySQLDB.
# Use it for options that only the server and MariaDB,
# you can put MariaDB-only options here
[mariadb]
# This group is only read by MySQLDB(=1) servers.
# If you use the same .cnf file for MariaDB of different versions,
# set the following options for options that older servers don't understand
[maradb]
```

Rysunek 66: mariadb – edycja pliku konfiguracyjnego

Restart usługi mariadb oraz instalacja serwera MySQL.



```
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 TO THE MAXIMUM EXTENT POSSIBLE.

It is recommended to log in as root and run this script using:
  mysql -u root -p

As root, you'll be prompted for the current root password, or for a new one if you've just installed MariaDB. 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):
+-----+
|          |
|          |
|          |
+-----+
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 go 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.

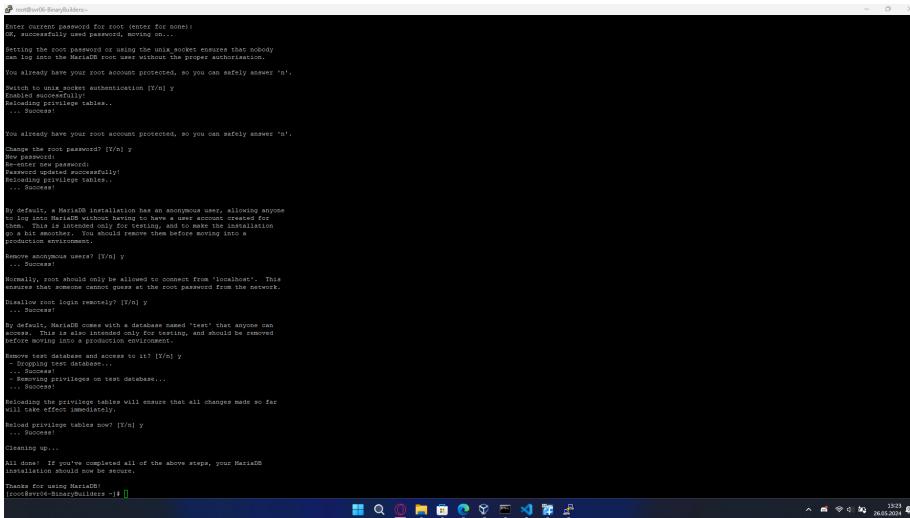
Remove test database and access to it? [Y/n] Y
- Dropping test database...
- Removing privileges on test database...
... Success!

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!
processefile-binaryBuilders -i
```

Rysunek 67: MySQL – instalacja część pierwsza



```
root@DevW8-Server:~# ./mysql_secure_installation
Enter current password for root (enter for none):
+-----+
|          |
|          |
|          |
+-----+
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 go 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!
processefile-binaryBuilders -i
```

Rysunek 68: MySQL – instalacja część druga

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

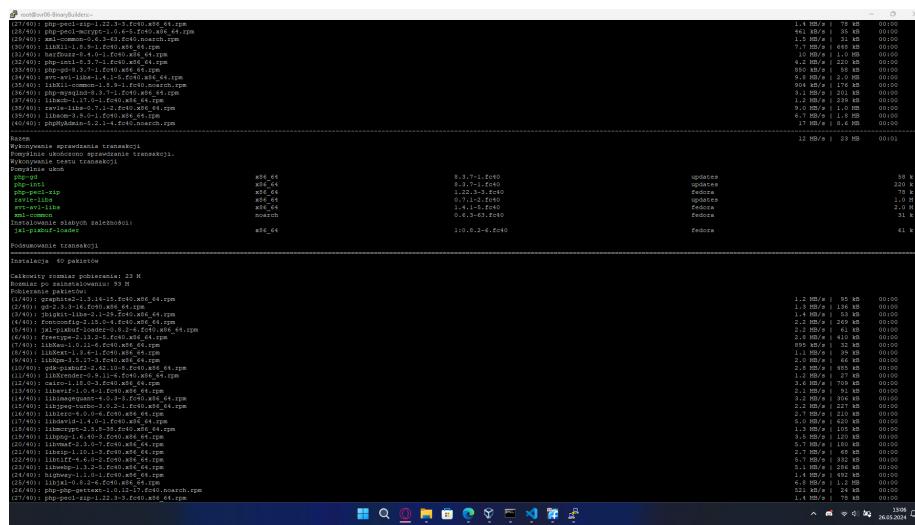
## 4.12 phpMyAdmin – instalacja i konfiguracja

Aby zainstalować phpMyAdmin należy użyć komendy:

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

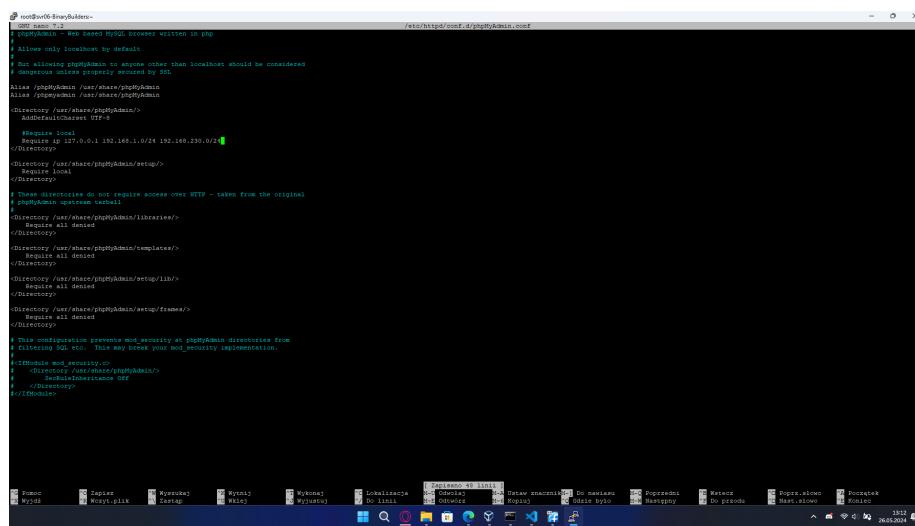
Po instalacji wykonuje kopię oryginalnego pliku konfiguracyjnego. Można to zrobić komendą:

```
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

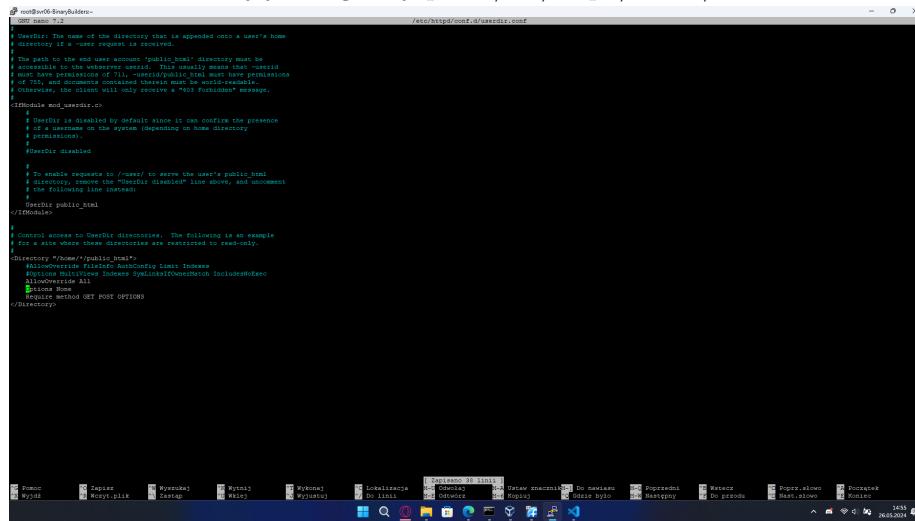
Test działania phpMyAdmin dostępny jest [tutaj](#).

## 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



```
# nano /etc/httpd/conf.d/userdir.conf

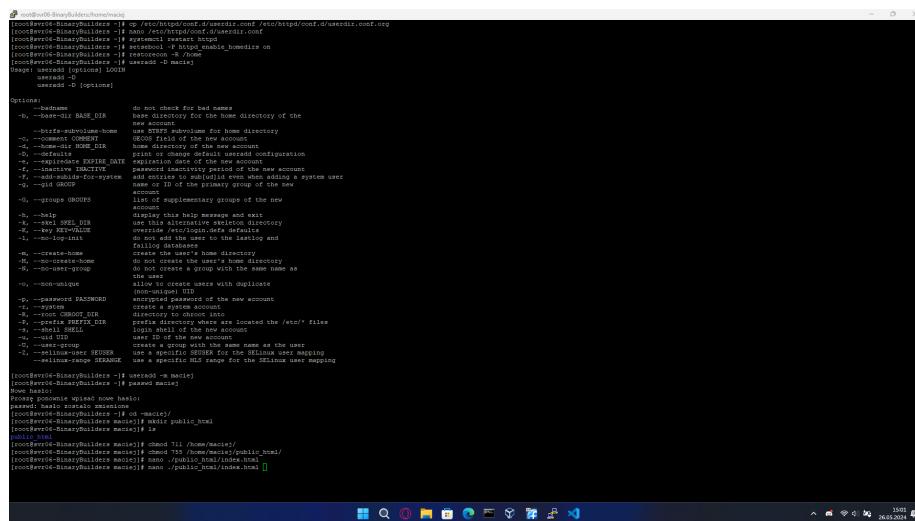
UserDir: The name of the directory that is appended onto a user's home
directory of a user's request is reversed.
For example, if a user requested "http://www.foobar.com/~userdir", then the
document directory would be "/home/userdir/www.foobar.com". The document
directory must be accessible to the webserver userid. This usually means that "userdir"
permissions must be the same as the webserver userid. If the permissions
of "userdir" are different than the webserver userid, then the client will receive
an "403 Forbidden" message.

#module mod_userdir.c
#
# UserDir is disabled by default since it can confuse the presence
# of files on the system (depending on home directory
# permissions).
#UserDir disabled
#
# To enable requests to /~userdir/ to serve the user's public_html
# directory, add the following line instead:
#UserDir public_html
#/ifModule

# Control access to UserDir directories. The following is an example
# for a site where these directories are restricted to read-only.
<Directory "/home/~public_html">
    AllowOverride None
    Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
    Order Allow,Deny
    Deny from All
    Allow from None
    Options None
    Require method GET POST OPTIONS
</Directory>
```

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

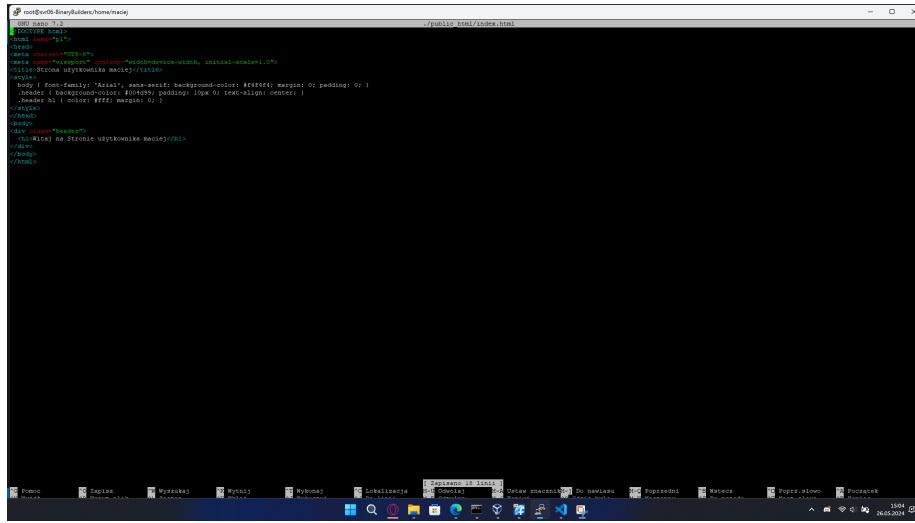
Teraz dodaję reguły SELinux, tworzę nowego użytkownika, oraz nadaję uprawnień odpowiednim katalogom, następnie tworzę stronę użytkownika maciej.



```
[root@v04-BinaryBuilders ~]# op /etc/httpd/conf.d/userdir.conf /etc/httpd/conf.d/userdir.conf.org
[root@v04-BinaryBuilders ~]# systemctl restart httpd
[root@v04-BinaryBuilders ~]# restorecon -R /home
[root@v04-BinaryBuilders ~]# restorecon -R /home/maciej
[maciej]: useradd (options) L001
        useradd -D (options)
Options)
--badname          do not check for bad name
--base             BASE_DIR      base directory for the home directory of the
                                new account
--beefs-subvolume-home   new account
--comment COMMENT     new account
--create-home      CREATE_HOME   create the user's home directory
--defaults         DEFAULTS      print or change default useradd configuration
--inactive INACTIVE   INACTIVE     password inactivity period of the new account
--add-groups-for-system   ADD_GROUPS   add entries to subpolicy entries defining a system user
--add-groups-for-user   ADD_GROUPS   add entries to subpolicy entries defining the new
                                account
--group GROUPS      GROUPS      supplementary group of the new
                                account
--help             HELP         print this help message and exit
--skel             SKELETON_DIR  use this alternative skeleton directory
--system           SYSTEM       create a system account
--no-log-init      NO_LOG_INIT  do not add the user to the logins and
                                faillog databases
--create-home      CREATE_HOME  create the user's home directory
--no-create-home    NO_CREATE_HOM do not create the user's home directory
--no-create-home-dir NO_CREATE_HOM_DIR do not create the user's home directory with the same name as
                                the user
--non-unique        NON_UNIQUE   create users with duplicate
                                (non-unique) GIDs
--password PASSWORD  PASSWORD     password for the new account
-i, --system         SYSTEM       create a system account
-p, --prefix PREFIX  PREFIX      prefix directory where are located the /etc/* files
-u, --uid UID        UID         user ID of the new account
-g, --gid GID        GID         group ID of the new account
--real-group REAL_GROUP  REAL_GROUP create a group with the same name as the user
--selinux-userkey SELINUX_KEY   SELINUX_KEY  SELinux user mapping
--selinux-range SELANGE  SELANGE    use a specific MLS range for the SELinux user mapping
[root@v04-BinaryBuilders ~]# useradd -D -u maciej
New hostid:
password: nowy hasło
password: hasło zostało zmienione
[root@v04-BinaryBuilders ~]# curl -I http://127.0.0.1/~maciej/
[root@v04-BinaryBuilders maciej]# mkdir public_html
[root@v04-BinaryBuilders maciej]# ls
[root@v04-BinaryBuilders maciej]# curl -I http://127.0.0.1/~maciej/
[root@v04-BinaryBuilders maciej]# curl -I http://127.0.0.1/~maciej/
[root@v04-BinaryBuilders maciej]# nano ./public_html/index.html
[root@v04-BinaryBuilders maciej]# nano ./public_html/index.html
```

Rysunek 72: Dodanie użytkownika maciej

Kod html tej strony



```
root@xen0-BinaryBuilders:~/home/maciej# id -u
0
root@xen0-BinaryBuilders:~/home/maciej# useradd --create-home --password maciej maciej
root@xen0-BinaryBuilders:~/home/maciej# id -u
1000
root@xen0-BinaryBuilders:~/home/maciej# usermod -aG users maciej
root@xen0-BinaryBuilders:~/home/maciej# nano ./public_html/index.html
root@xen0-BinaryBuilders:~/home/maciej# cat ./public_html/index.html
<!DOCTYPE HTML>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>
<h1>Kod html tej strony</h1>
<h2>Witaj na stronie użytkownika maciej</h2>
</body>
</html>
```

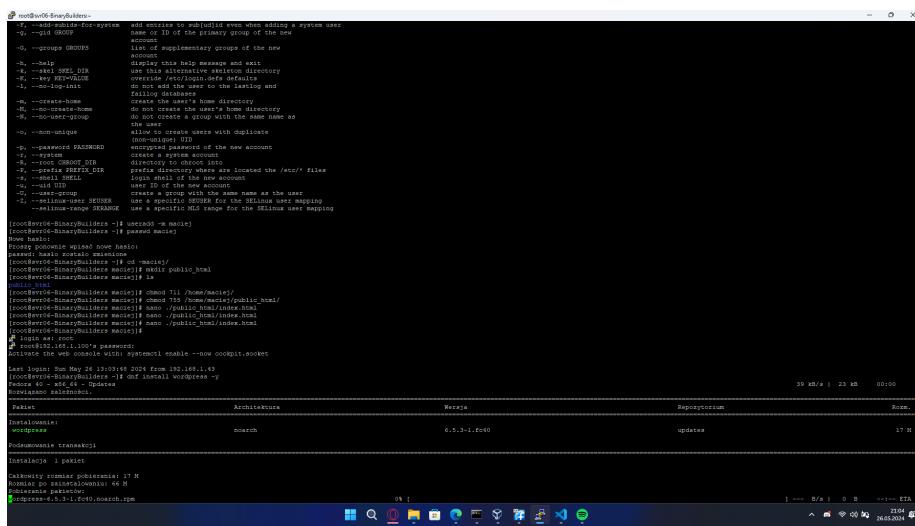
Rysunek 73: Stworzenie strony użytkownika maciej

Test działania dostępny jest [tutaj](#).

#### 4.14 WordPress – instalacja i konfiguracja

Aby zainstalować WordPress'a należy wykonać polecenie:

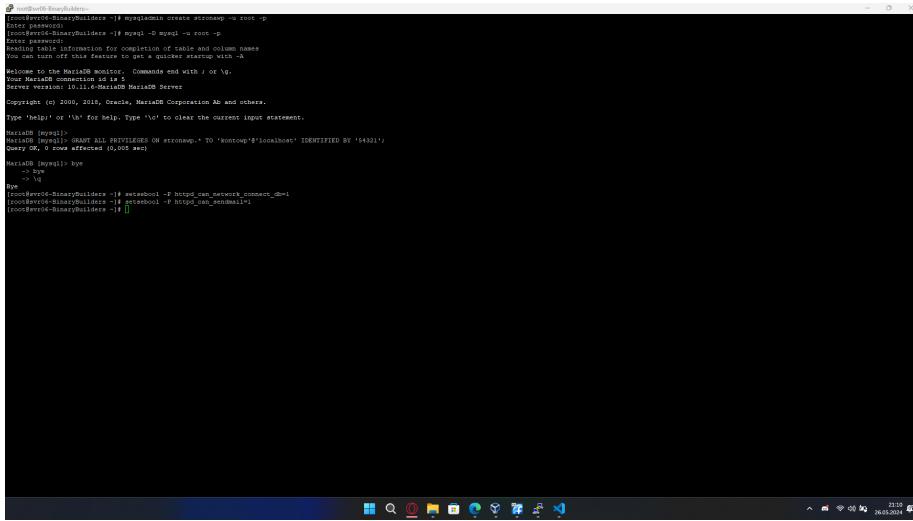
```
sudo dnf install wordpress -y
```



```
root@xen0-BinaryBuilders:~# useradd --create-home --password maciej maciej
root@xen0-BinaryBuilders:~# id -u
0
root@xen0-BinaryBuilders:~# echo 'maciej' | passwd --stdin maciej
root@xen0-BinaryBuilders:~# ls -l /var/www/
total 0
root@xen0-BinaryBuilders:~# cd /var/www/
root@xen0-BinaryBuilders:~/www# rm -rf *
root@xen0-BinaryBuilders:~/www# cp -r /var/www/public_html/* .
root@xen0-BinaryBuilders:~/www# ls
index.html
root@xen0-BinaryBuilders:~/www# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12, thread id is 41
mysql> create database maciej;
Query OK, 0 rows affected (0.000 sec)
mysql> use maciej;
Database changed
mysql> source /var/www/maciej/wp-config-sample.php
source: line 1: syntax error near unexpected token `='
source: line 1: `define('DB_NAME', 'maciej');'
mysql> exit
Bye
root@xen0-BinaryBuilders:~# curl 192.168.1.100:80
Retrieving the web page with: curl http://192.168.1.100/
[...]
root@xen0-BinaryBuilders:~# curl 192.168.1.100:80
Retrieving the web page with: curl http://192.168.1.100/
[...]
root@xen0-BinaryBuilders:~# curl 192.168.1.100:80
Retrieving the web page with: curl http://192.168.1.100/
[...]
```

Rysunek 74: WordPress – instalacja

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



```
[root@Dev04-BinaryBuilders ~]# mysqladmin create strongp -u root -p
[root@Dev04-BinaryBuilders ~]# mysql -D strongp -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 [strongp]>
MariaDB [strongp]> GRANT ALL PRIVILEGES ON strongp.* TO 'kontowp'@'localhost' IDENTIFIED BY 'kontowp123';
MariaDB [strongp]> FLUSH PRIVILEGES;
MariaDB [strongp]> \q
Bye
[root@Dev04-BinaryBuilders ~]# netcatool -P httpd_0m_n3wv3rK_c0nnect_0M=1
[root@Dev04-BinaryBuilders ~]# netcatool -P httpd_0m_s3ndemai1
[root@Dev04-BinaryBuilders ~]#
```

Rysunek 75: Stworzenie bazy danych dla WordPress'a

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

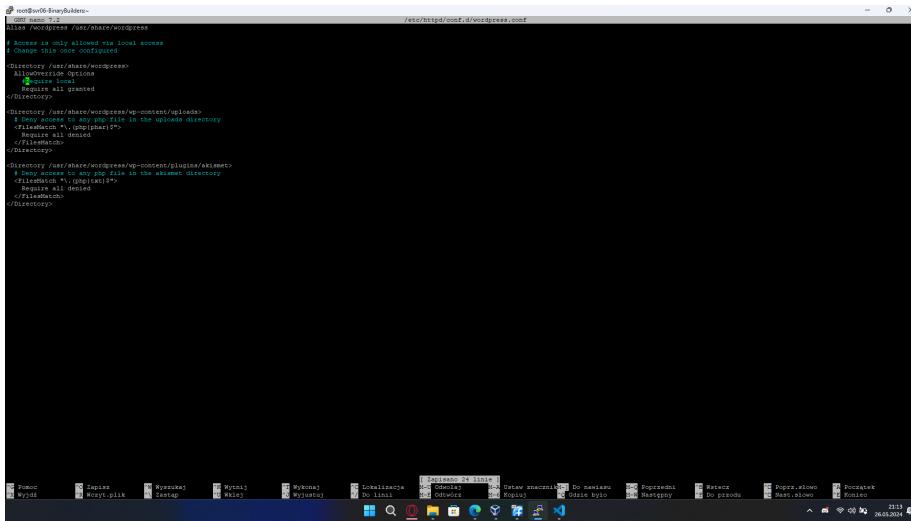
```
Alias /wordpress /usr/share/wordpress

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

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

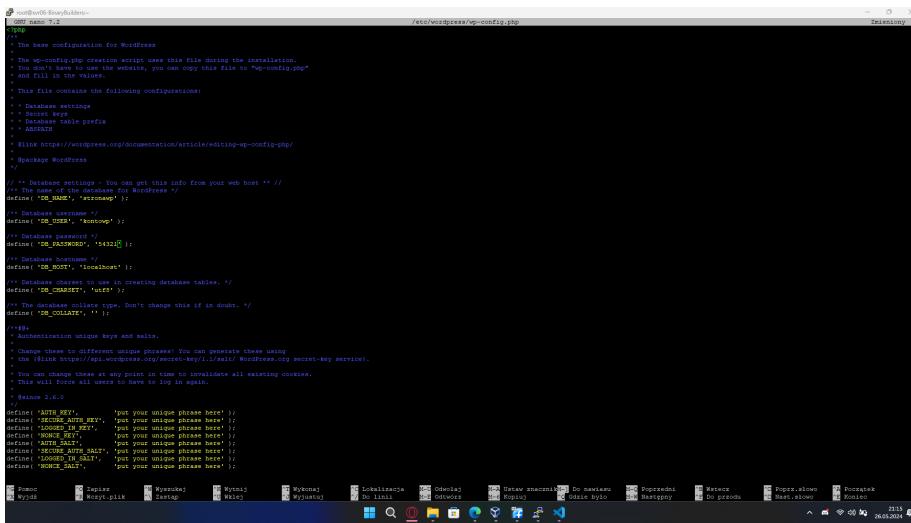
<Directory /usr/share/wordpress/wp-content/uploads>
    # Deny access to any php file in the uploads directory
    <FilesMatch "\.(php|phar)$">
        Require all denied
    </FilesMatch>
</Directory>

<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.

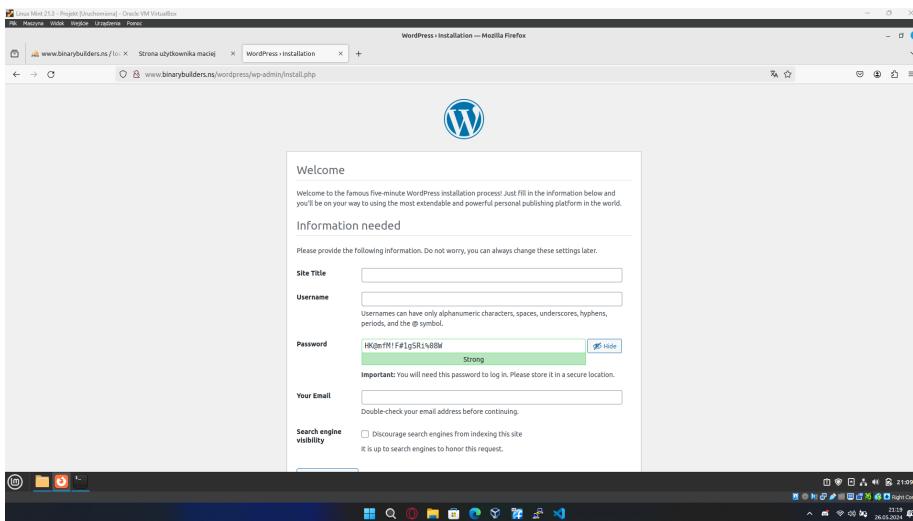


Rysunek 77: Edycja pliku /etc/wordpress/wp-config.php

W kolejnym kroku należy wykonać następujące polecenia:

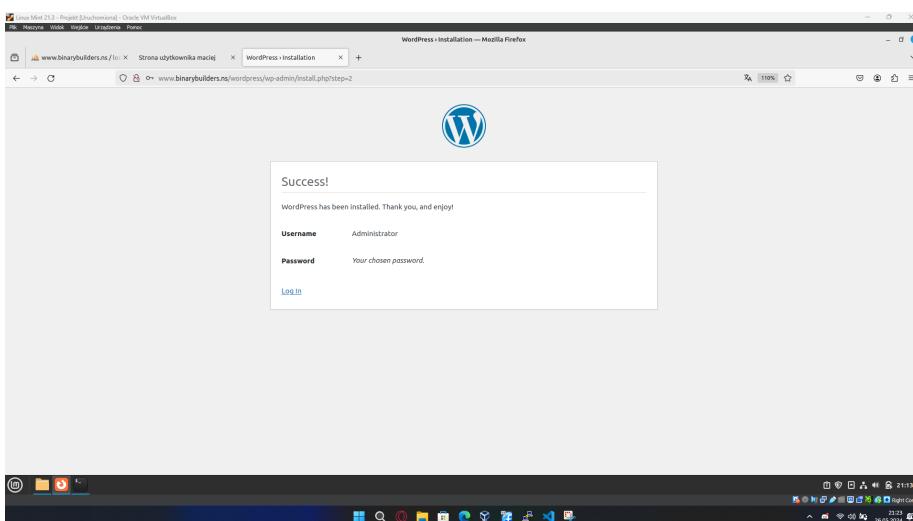
```
sudo setsebool -P httpd_can_network_connect_db=1
sudo setsebool -P httpd_can_sendmail=1
sudo systemctl restart httpd
```

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



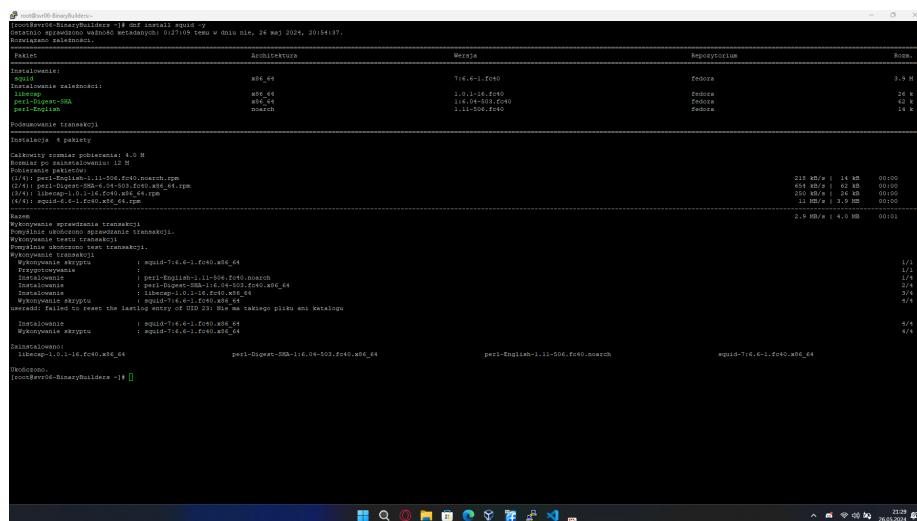
Rysunek 79: Instalacja WordPress – sukces

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

## 4.15 Proxy – instalacja i konfiguracja

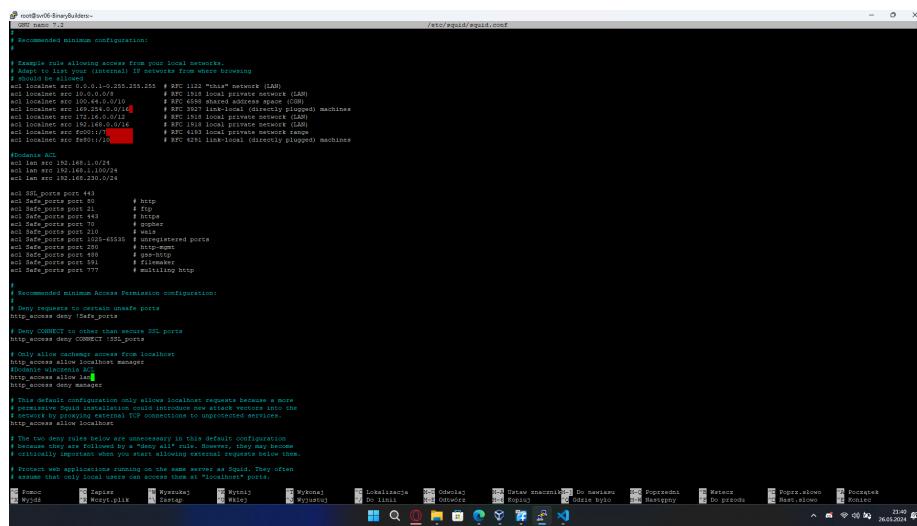
Aby zainstalować serwer proxy należy wydać polecenie:

```
sudo dnf install squid -y
```



Rysunek 80: Proxy – instalacja

W kolejnym kroku trzeba dodać do pliku /etc/squid/squid.conf zawartość jak w dwóch kolejnych zdjęciach.



Rysunek 81: Proxy – konfiguracja część pierwsza

Rysunek 82: Proxy – konfiguracja część druga

Po skonfigurowaniu tego pliku należy wydać następujące komendy:

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

Test proxy jest dostępny [tutaj](#).

## 4.16 SMTP – instalacja i konfiguracja

Komenda do instalacji SMTP:

```
sudo dnf install postfix -y
```

Rysunek 83: Proxy – konfiguracja część druga

W kolejnym kroku trzeba dodać do pliku /etc/postfix/main.cf zawartość tak jak na dwóch kolejnych zdjęciach.

Rysunek 84: Konfiguracja /etc/postfix/main.cf – część pierwsza

Rysunek 85: Konfiguracja /etc/postfix/main.cf – część druga

Po konfiguracji trzeba wykonać następujące komendy:

```
systemctl enable --now postfix  
firewall-cmd --add-service=smtp --permanent  
firewall-cmd --reload
```

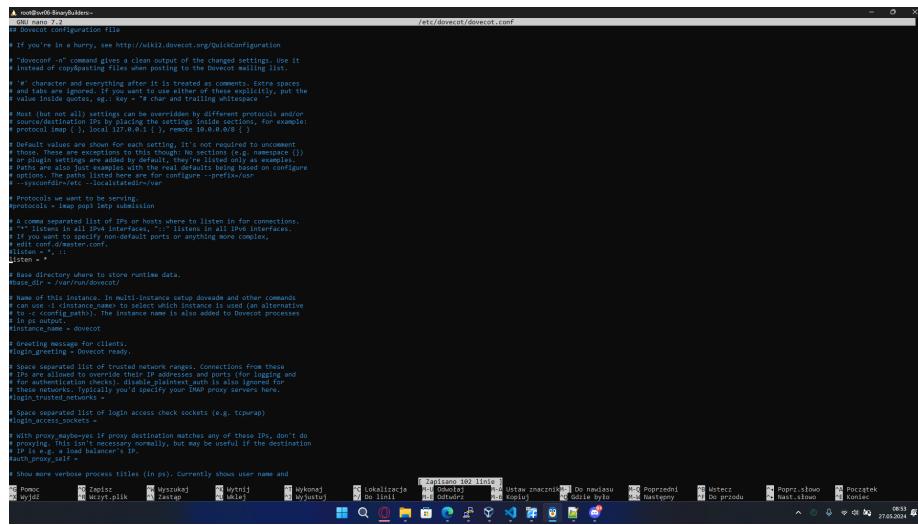
Test poczty jest dostępny [tutaj](#).

## 4.17 POP-IMAP – instalacja i konfiguracja

Aby wdrożyć POP/IMAP należy użyć polecenia:

```
sudo dnf install dovecot -y
```

Po zainstalowaniu wymaganych pakietów wykonuje kopie tego pliku /etc/dovecot/dovecot.conf. Następnie modyfikuje ten plik w następujący sposób:



```
# Dovecot 2.2.25-1.el7.x86_64
# Dovecot configuration file
# If you're in a hurry, see http://wiki.dovecot.org/QuickConfiguration

#doveconf -n command gives a clean output of the changed settings, use it
# to see what changes were made. To see all settings, use dovecot -n.
# '!' character and everything after it is treated as comment. Extra spaces
# and tabs are ignored, if you want to use either of those explicitly, put the
# value inside quotes, e.g.: key = "    " or trailing whitespace
# Value can be a string, a number, a boolean (yes/no), a list of values or
# a source/destination map by placing the settings inside sections, for example:
# protocol {imap { local 127.0.0.1; }, remote 10.0.0.0/8; }

# Default values are shown for each setting, it's not required to uncomment
# them. If you want to use a different value, just remove the comment (space(s))
# or plugin settings are added by default, they're listed only as examples.
# If you want to use a different value, just remove the comment (space(s)) and
# edit the value. You can also add your own settings to the end of the file. To
# options, the paths listed here are for config -prefix /var
# /etc/dovecot/dovecot.conf /var
# /etc/dovecot/auth /var
# /etc/dovecot/master /var
# /etc/dovecot/ssl /var
# /etc/dovecot/imap /var
# /etc/dovecot/imap /var
# A comma separated list of IP:s or hosts where to listen in for connections.
# It listens in all IPv4 interfaces, '*' listens in all IPv6 interfaces.
# If you want to bind to specific ports or anything more complex,
# edit conf.d/master.conf.
#listen = *:143

# Base directory where to store runtime data.
base_dir = /var/run/dovecot

# Name of this instance. In multi-instance setup dovecot and other command
# line options can be prepended with instance name. This is also used as alternative
# to < config_path>. The instance name is also added to Dovecot processes
# instance_name = dovecot

# Greeting message for clients.
log_greeting = dovecot ready.

# Space separated list of allowed network ranges. Connections from those
# IPs are allowed to override their IP addresses and port (for logging and
# for authentication). disable_plaintext_auth is also ignored for
# these clients. You might want to do this if you'd configure your IMAP proxy servers here.
#plugin_trusted_networks = 192.168.1.0/24

# Space separated list of login access check sockets (e.g. tcpwrap)
#auth_access = proxy

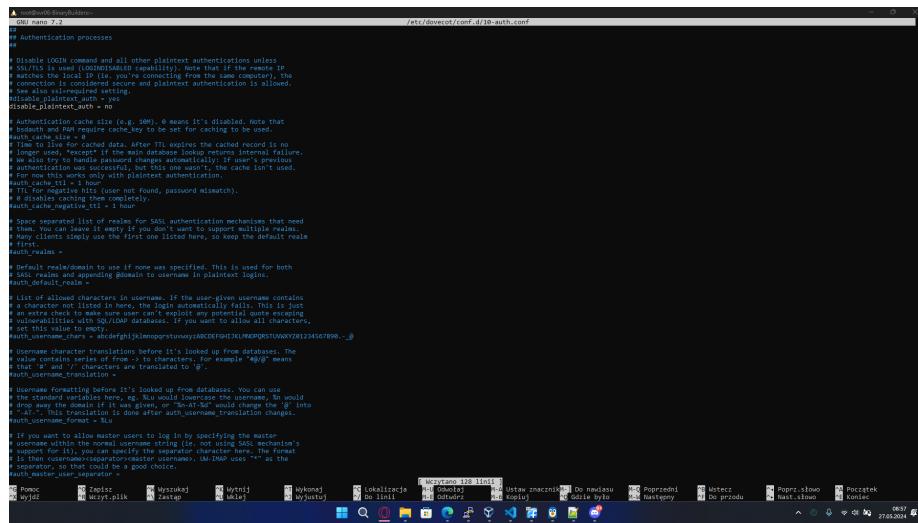
# If a client connects to proxy destination matches any of these IP:s, don't do
# proxying. This isn't necessary normally, but may be useful if the destination
# IP is e.g. a load balancer's IP.
#auth_proxy_self = 192.168.1.1

# Show more verbose process titles (in ps). Currently shows user name and
#pidfile = /var/run/dovecot.pid

# Domains
#mydd = Zapisz Wyślij Zestaw Wytnij Wykonaj Lokalizacja Od lini do linii Otwórz Ustaw znacznik Kopiu Poprzedni Następny Wyszczególn Do przodu Nast. słowo Po prawej Koniec
#myplik = Wyślij Zapisz Zestaw Wytnij Wykonaj Lokalizacja Od lini do linii Otwórz Ustaw znacznik Kopiu Poprzedni Następny Wyszczególn Do przodu Nast. słowo Po prawej Koniec
#myzapisz = Wyślij Zapisz Zestaw Wytnij Wykonaj Lokalizacja Od lini do linii Otwórz Ustaw znacznik Kopiu Poprzedni Następny Wyszczególn Do przodu Nast. słowo Po prawej Koniec
#myzestaw = Wyślij Zapisz Wytnij Wykonaj Lokalizacja Od lini do linii Otwórz Ustaw znacznik Kopiu Poprzedni Następny Wyszczególn Do przodu Nast. słowo Po prawej Koniec
#mywytnij = Wyślij Zapisz Zestaw Wykonaj Lokalizacja Od lini do linii Otwórz Ustaw znacznik Kopiu Poprzedni Następny Wyszczególn Do przodu Nast. słowo Po prawej Koniec
#mywykonaj = Wyślij Zapisz Zestaw Wytnij Lokalizacja Od lini do linii Otwórz Ustaw znacznik Kopiu Poprzedni Następny Wyszczególn Do przodu Nast. słowo Po prawej Koniec
#mylokalizacja = Wyślij Zapisz Zestaw Wytnij Wykonaj Od lini do linii Otwórz Ustaw znacznik Kopiu Poprzedni Następny Wyszczególn Do przodu Nast. słowo Po prawej Koniec
#myodliniidoLINI = Wyślij Zapisz Zestaw Wytnij Wykonaj Lokalizacja Otwórz Ustaw znacznik Kopiu Poprzedni Następny Wyszczególn Do przodu Nast. słowo Po prawej Koniec
#myotwórz = Wyślij Zapisz Zestaw Wytnij Wykonaj Lokalizacja Od lini do linii Ustaw znacznik Kopiu Poprzedni Następny Wyszczególn Do przodu Nast. słowo Po prawej Koniec
#myustawZNACZNIK = Wyślij Zapisz Zestaw Wytnij Wykonaj Lokalizacja Od lini do linii Otwórz Kopiu Poprzedni Następny Wyszczególn Do przodu Nast. słowo Po prawej Koniec
#mykopiu = Wyślij Zapisz Zestaw Wytnij Wykonaj Lokalizacja Od lini do linii Otwórz Ustaw znacznik Poprzedni Następny Wyszczególn Do przodu Nast. słowo Po prawej Koniec
#mypoprzedni = Wyślij Zapisz Zestaw Wytnij Wykonaj Lokalizacja Od lini do linii Otwórz Ustaw znacznik Kopiu Następny Wyszczególn Do przodu Nast. słowo Po prawej Koniec
#mynastepny = Wyślij Zapisz Zestaw Wytnij Wykonaj Lokalizacja Od lini do linii Otwórz Ustaw znacznik Kopiu Poprzedni Wyszczególn Do przodu Nast. słowo Po prawej Koniec
#mywyszczególn = Wyślij Zapisz Zestaw Wytnij Wykonaj Lokalizacja Od lini do linii Otwórz Ustaw znacznik Kopiu Poprzedni Następny Wyszczególn Do przodu Nast. słowo Po prawej Koniec
#mydoPRZODU = Wyślij Zapisz Zestaw Wytnij Wykonaj Lokalizacja Od lini do linii Otwórz Ustaw znacznik Kopiu Poprzedni Następny Wyszczególn Do przodu Nast. słowo Po prawej Koniec
#myNAST.ŚLÓW = Wyślij Zapisz Zestaw Wytnij Wykonaj Lokalizacja Od lini do linii Otwórz Ustaw znacznik Kopiu Poprzedni Następny Wyszczególn Do przodu Nast. słowo Po prawej Koniec
#myPODPRZODU = Wyślij Zapisz Zestaw Wytnij Wykonaj Lokalizacja Od lini do linii Otwórz Ustaw znacznik Kopiu Poprzedni Następny Wyszczególn Do przodu Nast. słowo Po prawej Koniec
#mykoniec = Wyślij Zapisz Zestaw Wytnij Wykonaj Lokalizacja Od lini do linii Otwórz Ustaw znacznik Kopiu Poprzedni Następny Wyszczególn Do przodu Nast. słowo Po prawej Koniec
```

Rysunek 86: Konfiguracja /etc/dovecot/dovecot.conf

W następnym kroku wykonuję kopię pliku /etc/dovecot/conf.d/10-auth.conf. Na dwóch następnych zdjęciach jest pokazane jak zmodyfikować ten plik.



```
# Dovecot 2.2.25-1.el7.x86_64
# Dovecot configuration file
# Authentication processes

# Disable LOGIN command and all other plaintext authentication unless
# SSL/TLS is used (LOGINDISABLED capability). Note that if the remote IP
# is not SSL/TLS, it's considered as being coming from the same computer.
# See also idle_timeout setting.
#disable_plaintext_auth = yes

# Authentication cache size (e.g. 10M). 0 means it's disabled. Note that
# SASL cache is separate and auth_cache is to be set for caching of users
# and their passwords. auth_cache_size = 0
auth_cache_size = 0

# Space separated list of realms for SASL authentication mechanisms that need
# realm information. It's used for SASL digest-md5 and similar mechanisms.
# Many clients simply use the first one listed here, so keep no defaults realms
#realm realms = 

# Default realm/domain to use if none was specified. This is used for both
# SASL realms and appending #domain to usernames in plaintext logins.

# List of allowed characters in username. If the user-given username contains
# a character not listed in here, the login automatically fails. This is just
# a list of characters, it does not mean that it's allowed to have multiple realms.
# When using SASL with SQL/IMAP databases, if you want to allow all characters,
# set this value to empty.
#auth_username_charset = abcdefghijklmopqrstuvwxyzABCDEFHJKLMOPQRSTUVWXYZ01234567890..._.

# Username character translations before it's looked up from databases. The
# value contains series of from -> to characters. For example "+@g" means
# that '@' is replaced with '+'. It's useful for IMAP users.
#auth_username_translation = +@g

# Usernames formatting before it's looked up from databases. You can use
# domain expansion here. If you want to expand domain part, you would
# drop away the domain if it was given, or "abc@.g" would change the ".g" into
# nothing. Format string is done after auth_username_translation changes.
#auth_username_format = %n

# If you want to allow master users to log in by specifying the master_
# username within the normal username string (i.e. not using SASL mechanism's
# own mechanism). This is useful for IMAP users. The separator character
# is then colon(:) separator of master username. IMAP uses ":" as the
# separator character.
#auth_master_user_separator = :
```

Rysunek 87: Konfiguracja /etc/dovecot/conf.d/10-auth.conf – część pierwsza

Rysunek 88: Konfiguracja /etc/dovecot/conf.d/10-auth.conf – część druga

W dalszej części konfiguruję plik /etc/dovecot/conf.d/10-mail.conf. W odpowiednim miejscu dopisuję linię:

```
mail_location = maildir:~/Maildir
```

```
# modprobe dovecot
GNU nano 7.2                                     /etc/dovecot/conf.d/10-mail.conf

# Mailbox locations and namespaces
#
# Location for users' mailboxes. The default is empty, which means that Dovecot
# will try to find the mailboxes automatically. This isn't very safe if the user
# doesn't yet have any mail, so you should explicitly tell Dovecot the full
# location.
#
# You can also use mbox, giving a path to the TMDB file (e.g. /var/mail/Mu)
# if that's enough. You'll also need to tell Dovecot where the other mailboxes are
# kept (this is called "root mail directory"), and it must be the first
# location given in the mail_location setting.
#
# There are a few special variables you can use, eg.:
#
# $user - username
# $domain - user part in user@domain, same as $u if there's no domain
# $maildir - user part in user@domain, empty if there's no domain
# $home - home directory
#
See doc/maildirvariables.txt for full list. Some examples:
#
mail_location = maildir:~/Maildir
mail_location = mbox:/mail/INBOX:/var/mail/Mu
mail_location = Mbox:/var/mail/krzysztof/mbox:/var/indexes/krzysztof/Maildir
#cdw/krzysztof/maillocation.txt
#
mail_location =
maildir:~/Maildir
#
# If you need to set multiple mailbox locations or want to change default
# namespace settings, you can do it by defining namespace sections.
#
# You can have private, shared and public namespaces. Private namespaces
# are for your own personal mail. Shared namespaces are for accessing other
# users' mail that you're allowed to. Public namespaces are for shared
# mailboxes that are managed by sysadmin. If you create any shared or public
# namespaces, you must define them in the mail_location section. This is
# because users can access all the shared mailboxes, assuming they have permissions
# to do so.
#
#namespace INBOX
#  type = public
#  type = private, shared or public
#  type = private
#
# Hierachy separator to use. You should use the same separator for all
# namespaces or some clients get confused. '/' is usually a good one.
# The separator however depends on the underlying mail storage format.
#separator =
#
# Prefix required to access this namespace. This needs to be different for
# different namespaces. For example "Public:".
#prefix =
#
# Prefix of the mailbox. This is in same format as
# $mail_location, which is also the default for it.
#location =
```

Rysunek 89: Konfiguracja /etc/dovecot/conf.d/10-mail.conf

W kolejnym kroku edytuję plik /etc/dovecot/conf.d/10-master.conf jak poniżej:

Rysunek 90: Konfiguracja /etc/dovecot/conf.d/10-master.conf

W kolejnym kroku edytuję plik /etc/dovecot/conf.d/10-ssl.conf. Dodaję linię:

```
ssl = yes
```

Rysunek 91: Konfiguracja /etc/dovecot/conf.d/10-ssl.conf

Po konfiguracji wykonuję następujące komendy:

```
systemctl enable --now dovecot  
firewall-cmd --add-service={pop3,imap} --permanent  
firewall-cmd --reload
```

Rysunek 92: Restart usługi Dovecot i dodanie do zapory sieciowej

Aby zainstalować klienta poczty na serwerze należy użyć komendy:

```
sudo dnf install mailx -y
```

Następnie tworzę plik /etc/profile.d/mail.sh i dodaję do niego następującą zawartość:

```
export MAIL=$HOME/Maildir
```

Następnie weryfikuje działanie powyższego skryptu:

Rysunek 93: Test zmiennej środowiskowej MAIL

Przykład użycia tekstuowego klienta poczty na serwerze:

Rysunek 94: mailx – przykład użycia

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

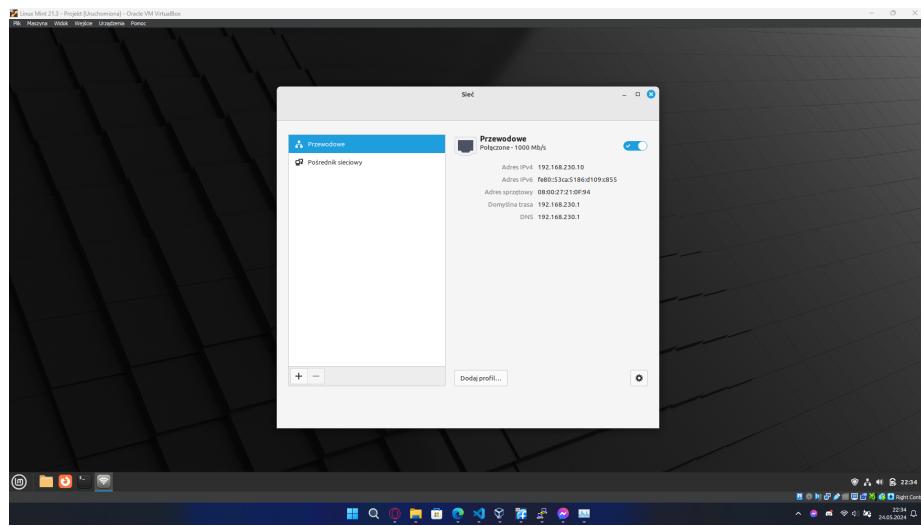
## 5 Testy działania wdrożonych usług

## 5.1 DNS

```
Linux Mint 21.3 - Bubblegum | Oracle VM VirtualBox  
Druk Masaż Wykroj Wyszukiwanie Terminal Pomoc  
maciej@maciej:~$ ping -c 2 www.BinaryBuilders.ns  
ping: www.BinaryBuilders.ns: Name or service not known  
maciej@maciej:~$ dig www.BinaryBuilders.ns  
;; <> DiG 9.18.18-Debian@0.22.04.2Ubuntu <>> www.BinaryBuilders.ns  
global options: +ndots  
; global options: +cd  
; Got answer from 192.168.230.1#53  
;+>HEADER<> opcode: QUERY, status: NOERROR, id: 65494  
; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1  
; OPT PSEUDOSECTION:  
; EDNS: version: 0, flags: udp; udp: 65494  
; QUESTION SECTION:  
www.BinaryBuilders.ns. IN A  
; ANSWER SECTION:  
www.BinaryBuilders.ns. 7120 IN A 192.168.230.1  
; Query time: 4 msec  
; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)  
; WHEN: Fri May 24 23:51:20 CEST 2024  
; MSG SIZE (rcvd): 69  
  
maciej@maciej:~$ dig www.BinaryBuilders.ns  
;; <> DiG 9.18.18-Debian@0.22.04.2Ubuntu <>> www.BinaryBuilders.ns  
global options: +cd  
; global options: +cd  
;+>HEADER<> opcode: QUERY, status: NOERROR, id: 47558  
; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1  
; OPT PSEUDOSECTION:  
; EDNS: version: 0, flags: udp; udp: 65494  
; QUESTION SECTION:  
www.BinaryBuilders.ns. IN A  
; ANSWER SECTION:  
www.BinaryBuilders.ns. 7131 IN A 192.168.230.1  
; Query time: 4 msec  
; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)  
; WHEN: Fri May 24 23:51:20 CEST 2024  
; MSG SIZE (rcvd): 69  
  
maciej@maciej:~$ ping -c 2 www.BinaryBuilders.ns  
PING www.BinaryBuilders.ns [192.168.230.1] 56(84) bytes of data.  
48 bytes from svr08.BinaryBuilder.BinaryBuilders.ns [192.168.230.1]: icmp seq=1 ttl=64 time=1.46 ms  
48 bytes from svr08.BinaryBuilder.BinaryBuilders.ns [192.168.230.1]: icmp_seq=2 ttl=64 time=1.26 ms  
2 packets transmitted, 2 received, 0% packet loss, time 1802ms  
rtt min/avg/max/mdev = 1.250/1.358/1.468/0.162 ms  
maciej@maciej:~$
```

Rysunek 95: Test DNS

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



Rysunek 96: Instalacja DHCP

## 5.2 DHCP

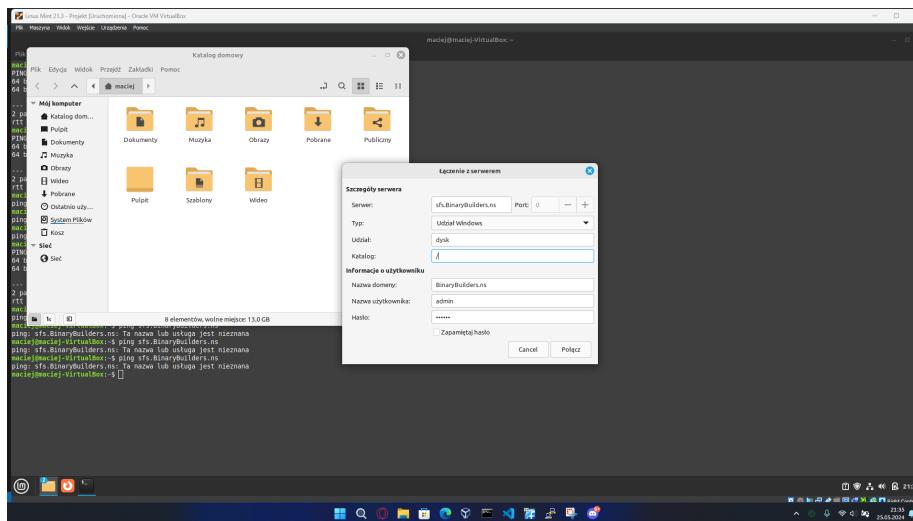
Jak widać na powyższym zdjęciu karta w systemie klienta ustawiona na sieć wewnętrzną dostała poprawny adres IP, adres bramy domyślnej i DNS. Na zdjęciu również widać że pula DHCP działa poprawnie.

### 5.3 Raid 5

Rysunek 97: Test automatycznego montowania partycji po ponownym uruchomieniu serwera

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

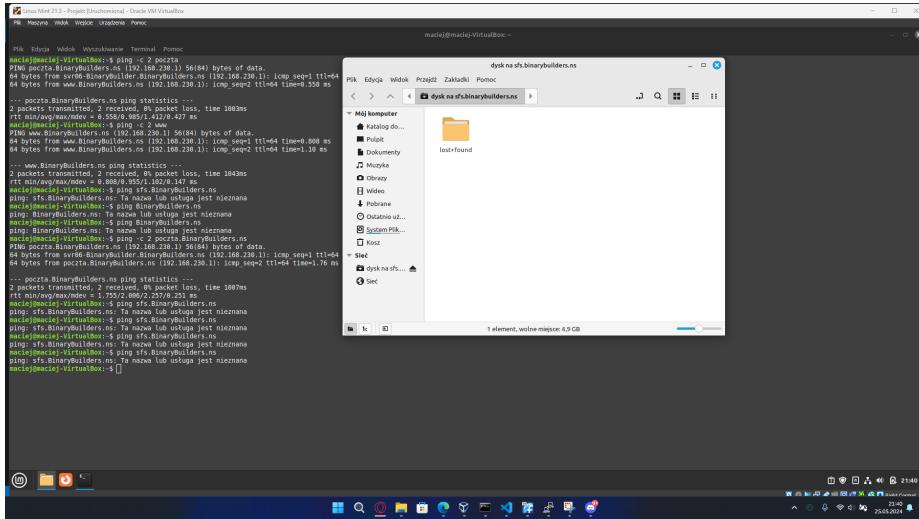
## 5.4 Samba



Rysunek 98: Samba – próba podłączenia się do udziału na serwerze

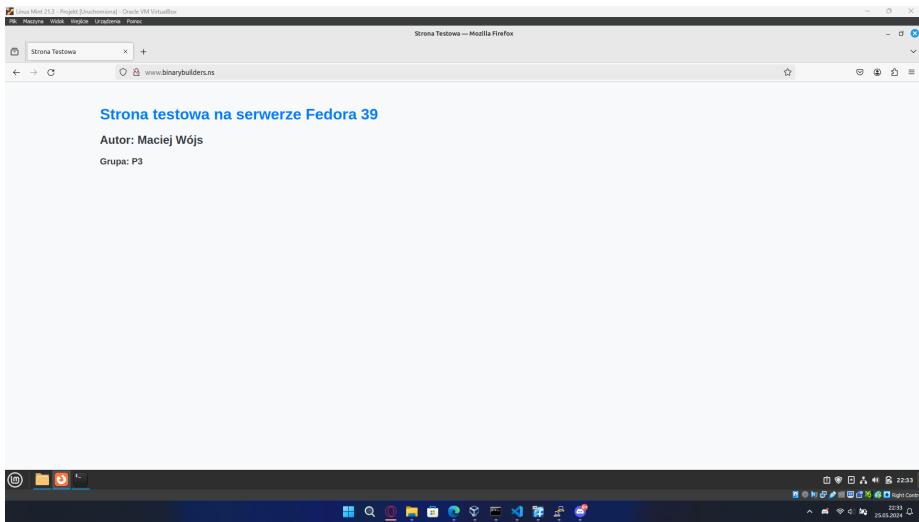
Aby podłączyć się z serwerem Samby należy otworzyć menadżer plików następnie otworzyć zakładkę Plik w lewym górnym rogu, następnie połącz z serwerem, kolejno w typie należy wybrać Udział Windows, ostatecznie należy

wypełnić wymagane dane. Przykładowa próba podłączenia powyżej, a efekt tego działania poniżej.



Rysunek 99: Samba – wynik poprzedniego kroku

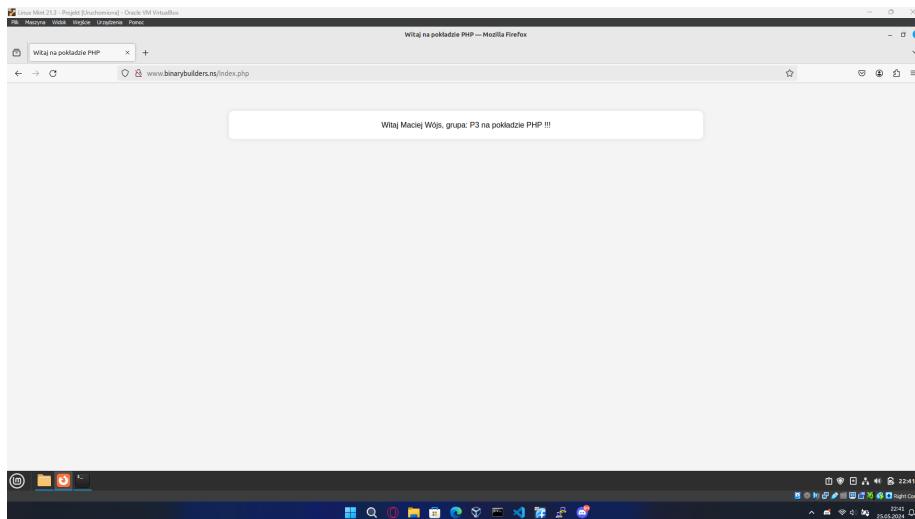
## 5.5 HTTP



Rysunek 100: Test działania serwera WWW

Jak widać na zdjęciu powyżej web serwer działa poprawnie.

## 5.6 PHP



Rysunek 101: Strona html + PHP

Jak można zauważyc strona wykorzystującą 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,
and you have privileges for all databases and users.
MySQL thread id: 10 [aborted]
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)

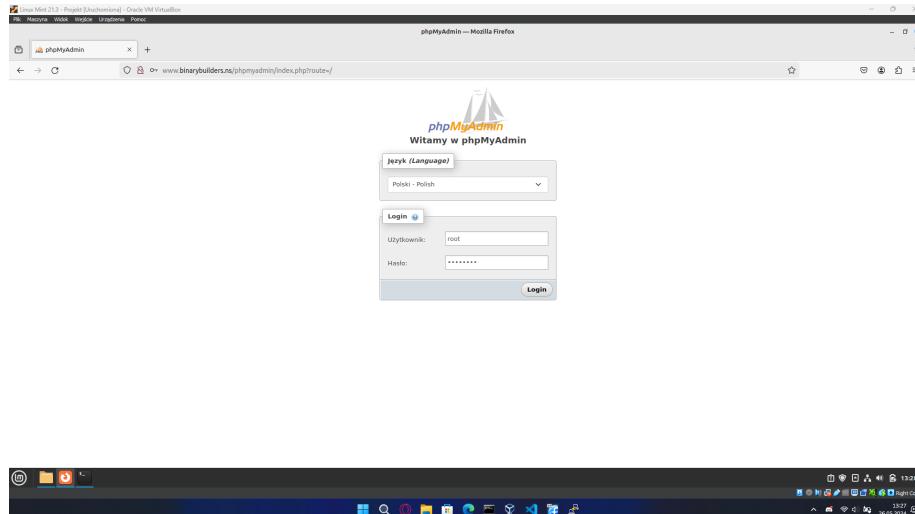
MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| performance_schema |
| sys |
+-----+
3 rows in set (0,001 sec)

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

Rysunek 102: Test usługi mariadb (MySQL)

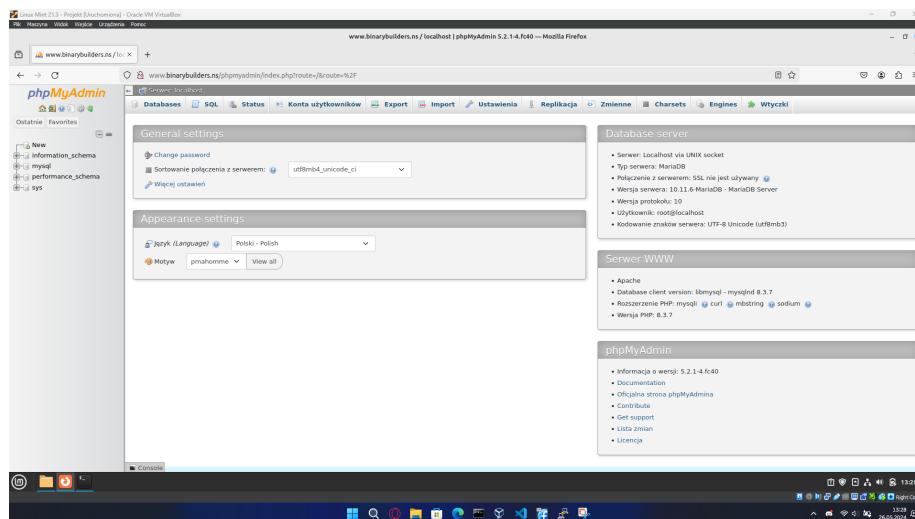
Jak widać połaczenie zz bazą danych działa. Następnym i ostatnim krokiem jest dodanie usługi MySQL do dozwolonych usług w zaporze ogniwowej.

## 5.8 phpMyAdmin



Rysunek 103: Test usługi phpMyAdmin – część pierwsza

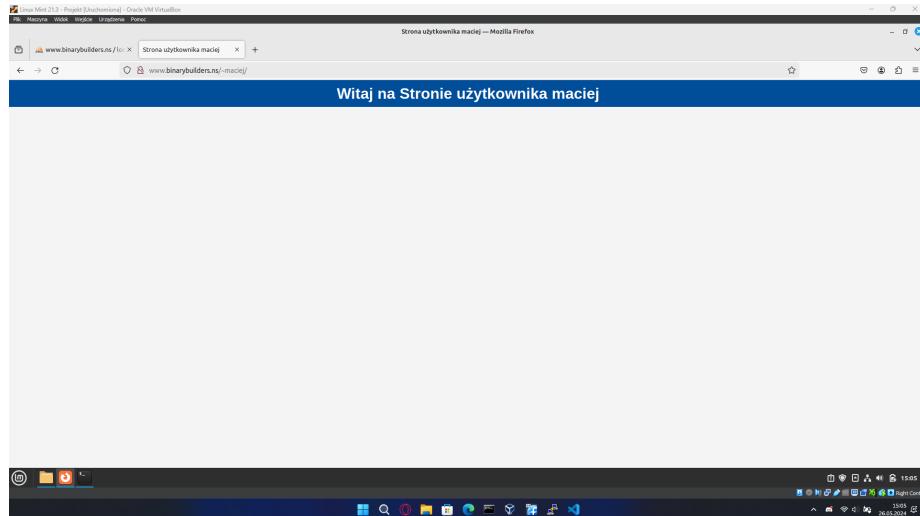
Pierwszym krokiem w testowaniu phpMyAdmin jest otworzenie strony serwera tej usługi w przeglądarce a następnie zalogowanie się na konto.



Rysunek 104: Test usługi phpMyAdmin – część druga

Jak widać po zalogowaniu mamy dostęp do administrowania bazami danych.

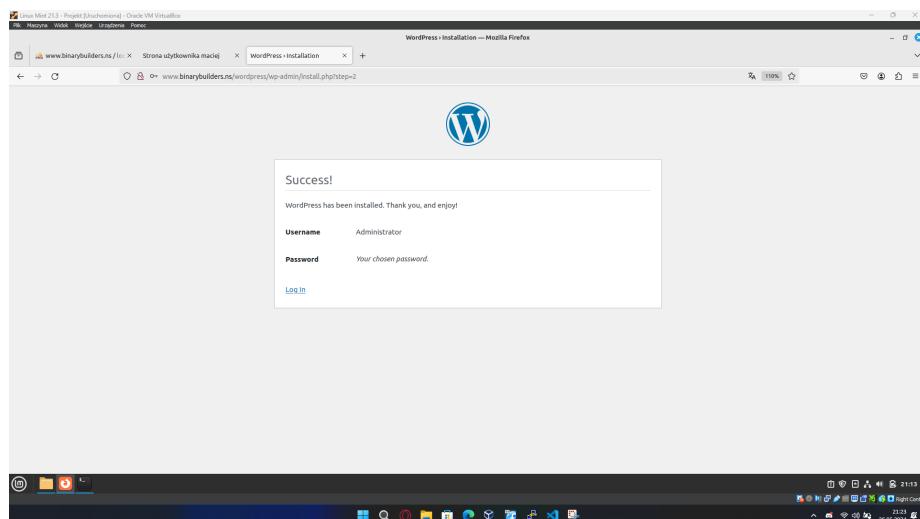
## 5.9 UserDir – serwer http



Rysunek 105: MySQL – instalacja część druga

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

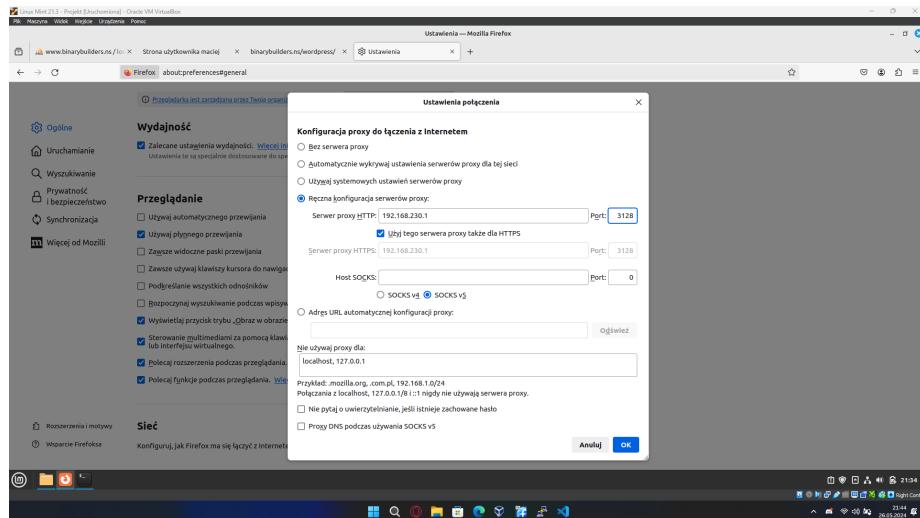
## 5.10 WordPress



Rysunek 106: Dashboard WordPress'a

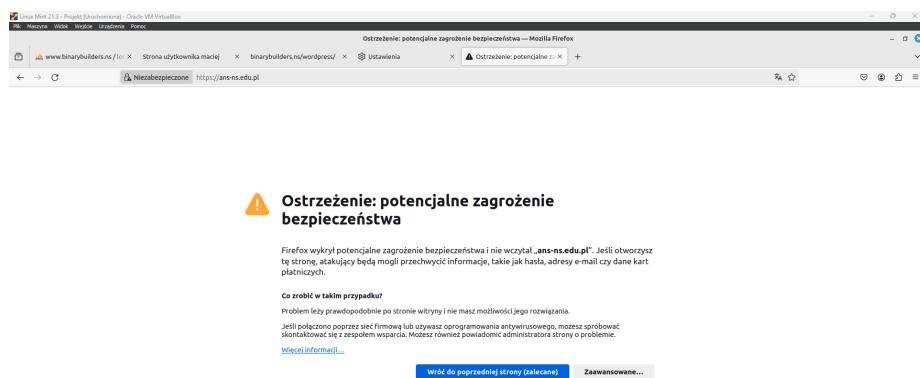
Jak widać powyżej po zalogowaniu się na konto widać dashboard WordPress'a, skąd można skonfigurować stronę internetową.

## 5.11 Proxy



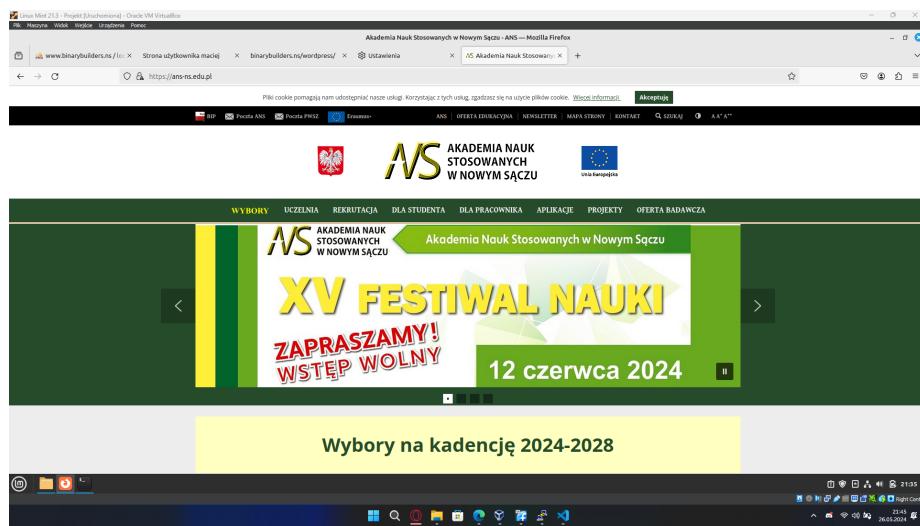
Rysunek 107: Proxy – ustawienie w FireFox

Konfiguracja proxy w przeglądarce FireFox.



Rysunek 108: Proxy – dostęp do strony ostrzeżenie

Próba podłączenia ze stroną **ANS**, skutkuje ostrzeżenie o zagrożeniu bezpieczeństwa.



Rysunek 109: Proxy – wynik strony po zignorowaniu ostrzerazenia

Widok strony po zignorowaniu ostrzeżenia.

Rysunek 110: Proxy – monitoring ruchu sieciowego z serwera

Jak widać na powyższym zrzucie ekranu proxy działa poprawnie. Na serwerze wyświetlane są strony z jakimi próbuje się połączyć klient. Przy obecnym ustawieniu proxy nie możliwa jest aktualizacja systemu. Rozwiązaniem tego problemu jest dodanie kilku linijek do /etc/environment. Limijki które trzeba dodać:

```
http_proxy="http://192.168.230.1:3128/"  
https_proxy="http://192.168.230.1:3128/"  
ftp_proxy="http://192.168.230.1:3128/"  
no_proxy="localhost,127.0.0.1,::1"
```

## 5.12 Poczta

```
Administrator:~ binarybuilders$ mail adm@binarybuilders.ns
mailadm@binarybuilders:~ $ mail adm@binarybuilders.ns
To: adm@binarybuilders.ns
Message saved

[preliminary] Envelope contains:
To: adm@binarybuilders.ns

Send this message [yes/no, empty: recompose]? yes
sendmail: /etc/mail/sendmail.cf, line 752: overriding earlier entry: mydestination=$myhostname, localhost.$mydomain, localhost
sendmail: warning: /etc/postfix/main.cf, line 752: overriding earlier entry: mydestination=$myhostname, localhost.$mydomain, localhost
postmap: warning: /home/admin/Maildir
postmap: warning: /home/admin/Maildir

adm@over90-BinaryBuilders:~ $
```

Rysunek 111: Wysłanie maila na serwerze

## 6 Kod skryptu BASH, oraz tablica crontab

FFFFFFFFFFFFFFFFFn

7 Wnioski

EEEEEEEEEEEEEEEE

## 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>.