

# **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
4.18 Dodatkowa usługa git (jako serwer) – instalacja i konfiguracja . . . . .	62

<b>5 Testy działania wdrożonych usług</b>	<b>65</b>
5.1 DNS . . . . .	65
5.2 DHCP . . . . .	66
5.3 Raid 5 . . . . .	66
5.4 Samba . . . . .	67
5.5 HTTP . . . . .	68
5.6 PHP . . . . .	68
5.7 MySQL . . . . .	69
5.8 phpMyAdmin . . . . .	70
5.9 UserDir – serwer http . . . . .	71
5.10 WordPress . . . . .	71
5.11 Proxy . . . . .	72
5.12 Poczta . . . . .	74
5.13 Git . . . . .	75
<b>6 Kod skryptu BASH, oraz tablica crontab</b>	<b>76</b>
<b>7 Wnioski</b>	<b>76</b>
<b>8 Literatura</b>	<b>77</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	git – konfiguracja . . . . .	65
96	Test DNS . . . . .	65
97	Instalacja DHCP . . . . .	66
98	Test automatycznego montowania . . . . .	66

99	Samba – próba podłączenia się do udziału na serwerze . . . . .	67
100	Samba – wynik poprzedniego kroku . . . . .	67
101	Test działania serwera WWW . . . . .	68
102	Strona html + PHP . . . . .	68
103	Test usługi mariadb (MySQL) . . . . .	69
104	Test usługi phpMyAdmin – część pierwsza . . . . .	70
105	Test usługi phpMyAdmin – część druga . . . . .	70
106	MySQL – instalacja część druga . . . . .	71
107	Dashboard WordPress'a . . . . .	71
108	Proxy – ustawienie w FireFox . . . . .	72
109	Proxy – dostęp do strony ostrzelenie . . . . .	72
110	Proxy – wynik strony po zignorowaniu ostrzelenia . . . . .	73
111	Proxy – monitoring ruchu sieciowego z serwera . . . . .	73
112	Wysłanie maila na serwerze . . . . .	74
113	Thunderbird - konfiguracja . . . . .	74
114	Wysłanie maila na serwerze . . . . .	75
115	git – konfiguracja . . . . .	75

# 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**

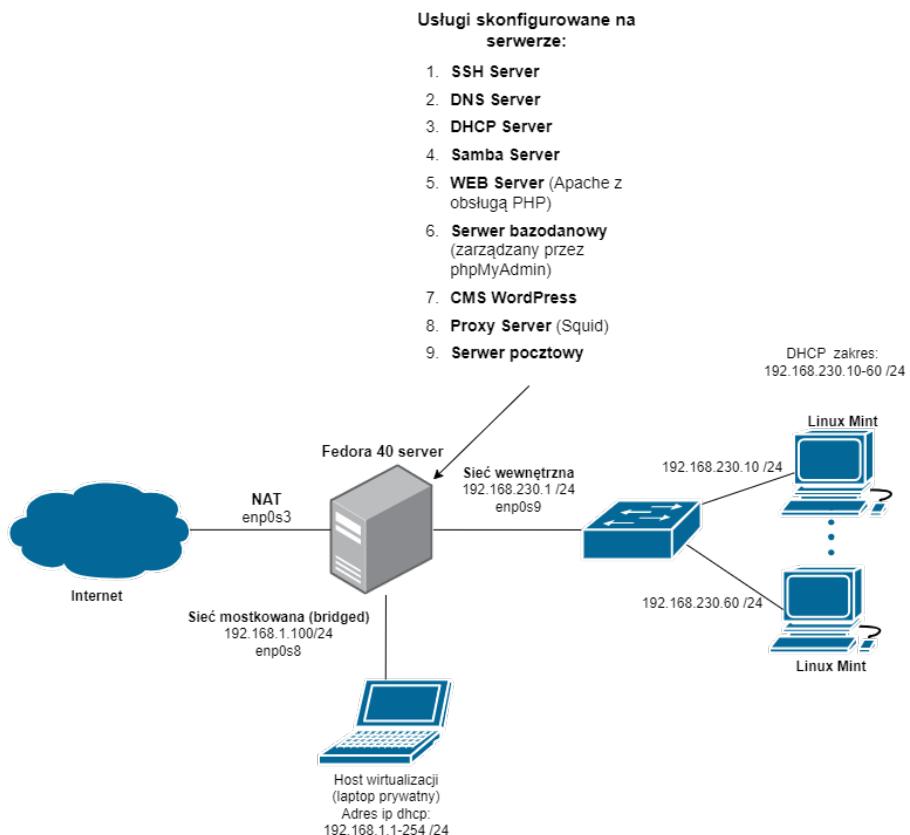
### **2.13.1 Skrypty Bash**

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

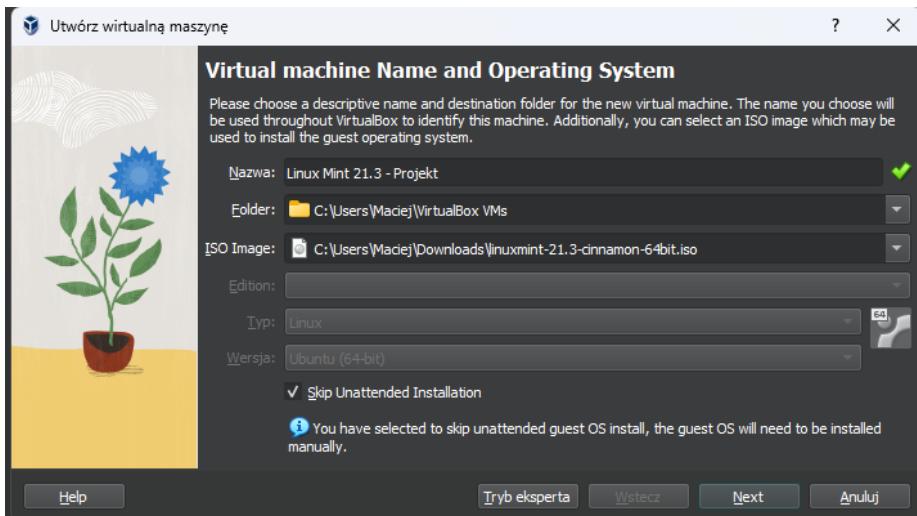


Rysunek 1: Schemat logiczny sieci

## 4 Procedury instalacyjne poszczególnych usług

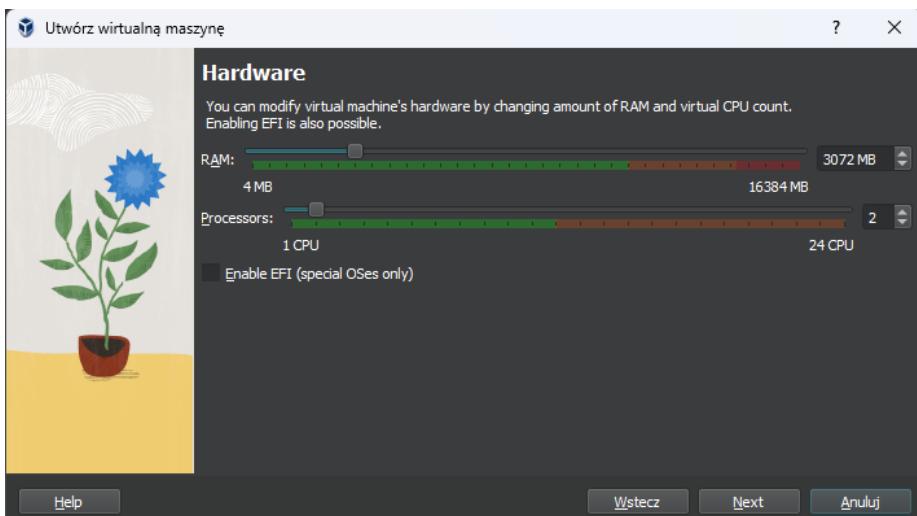
### 4.1 Instalacja systemu klienta – Linux Mint

#### 4.1.1 Proces instalacji



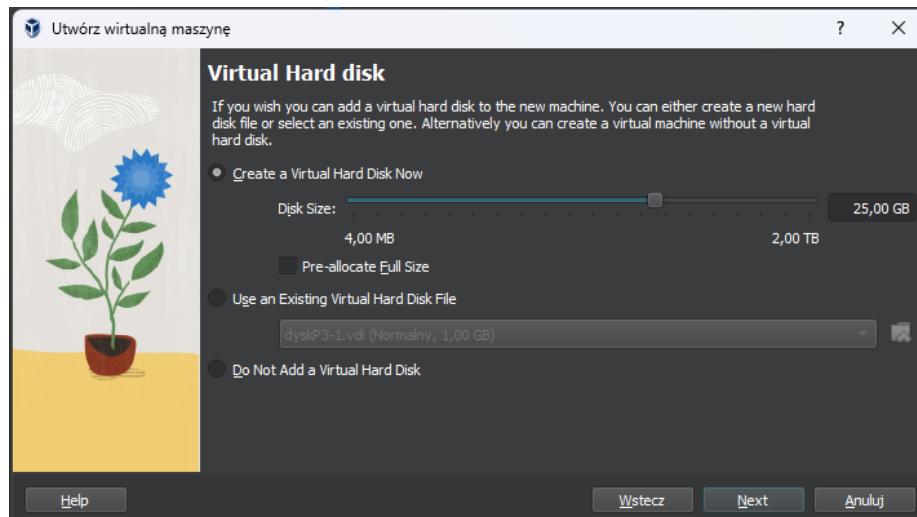
Rysunek 2: Tworzenie nowej maszyny wirtualnej. Ustawienia nazwy, lokalizacji dysku oraz wybór pliku ISO systemu operacyjnego.

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.



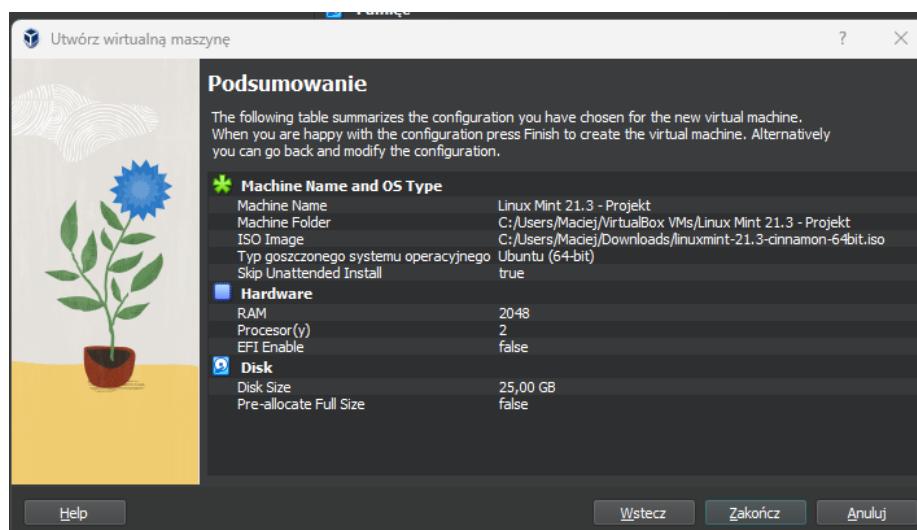
Rysunek 3: Przydzielanie zasobów maszynie wirtualnej, takich jak pamięć RAM i procesor.

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



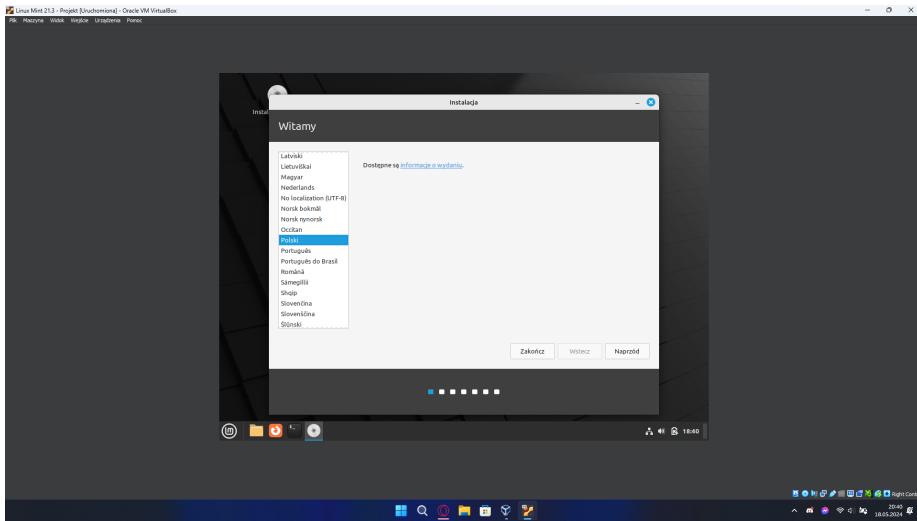
Rysunek 4: Określenie rozmiaru dysku wirtualnego.

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



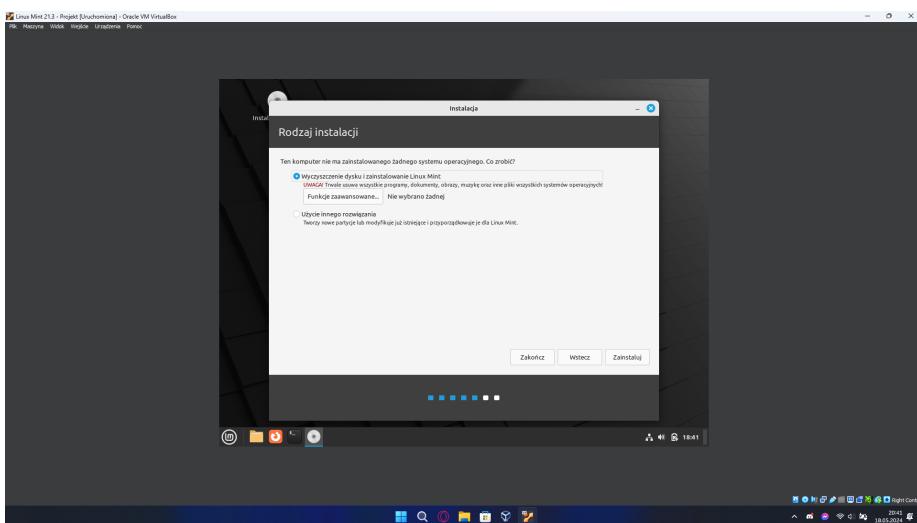
Rysunek 5: Podsumowanie konfiguracji maszyny wirtualnej przed rozpoczęciem instalacji systemu.

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



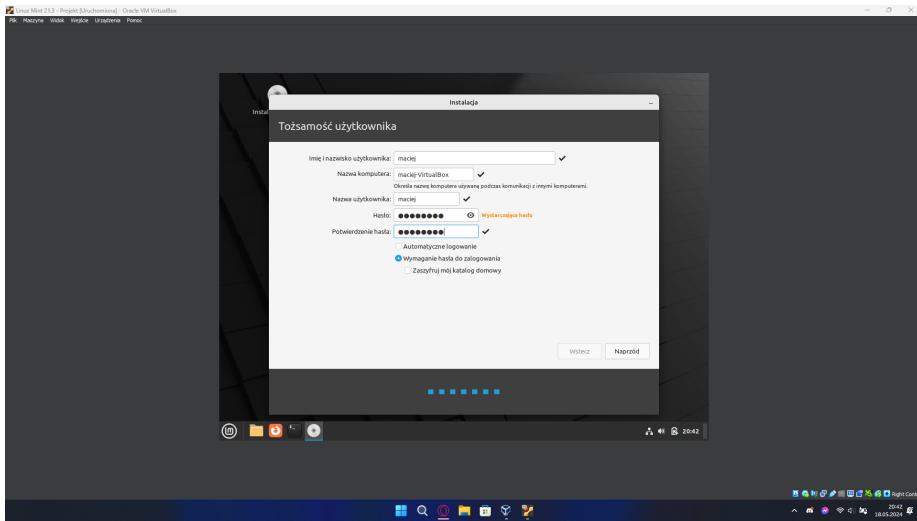
Rysunek 6: Rozpoczęcie instalacji Linux Mint – wybór języka instalacji.

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



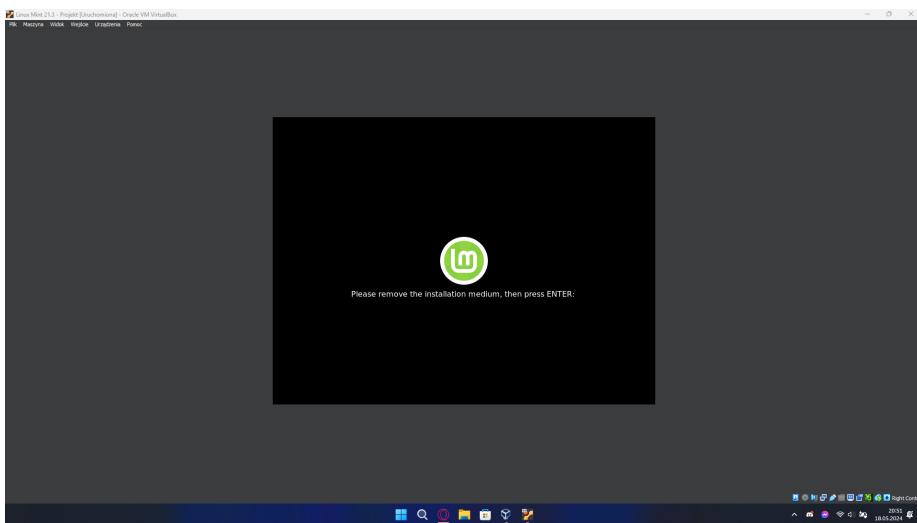
Rysunek 7: Wybór trybu instalacji na dysku twardym.

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.

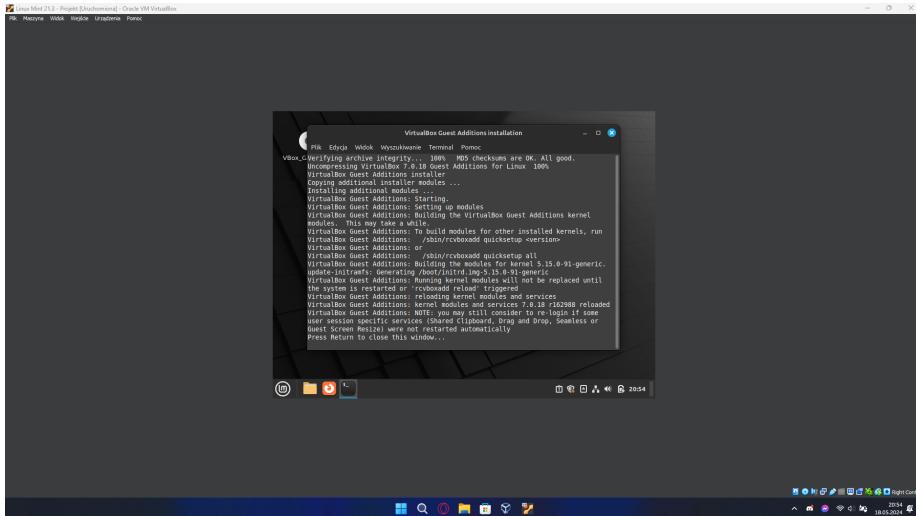
Kolejnym krokiem jest utworzenie konta użytkownika, wprowadzenie nazwy użytkownika, hasła oraz nazwy komputera.



Rysunek 9: Zakończenie instalacji systemu Linux Mint.

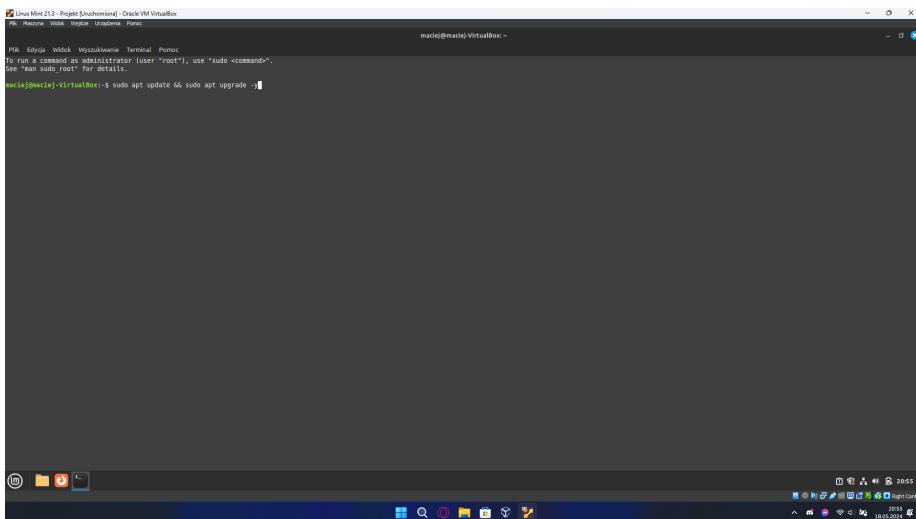
Wyświetlony zostaje monit z prośbą o usunięcie nośnika instalacyjnego. Po zakończeniu instalacji system wyświetla ekran informujący o pomyślnym zakończeniu procesu.

#### 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.

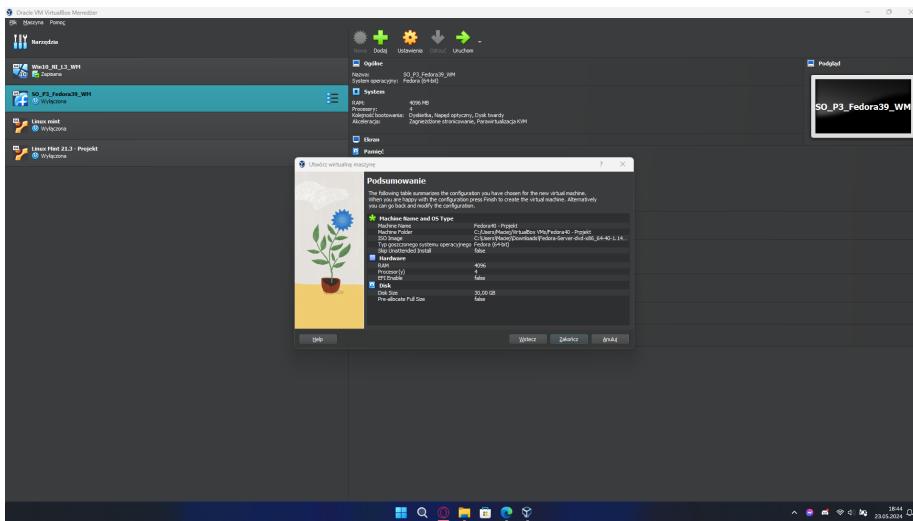


Rysunek 11: Aktualizacja pakietów systemowych.

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

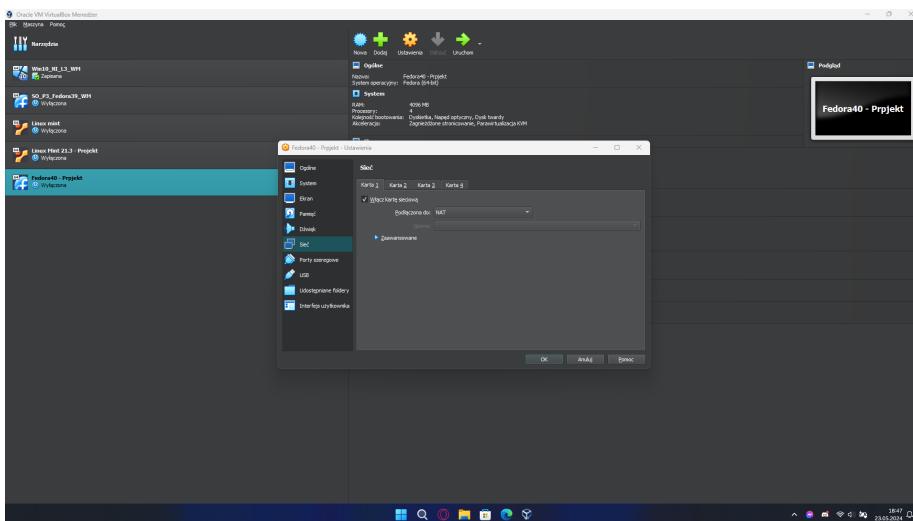
## 4.2 Instalacja serwera – Fedora 40

### 4.2.1 Proces instalacji



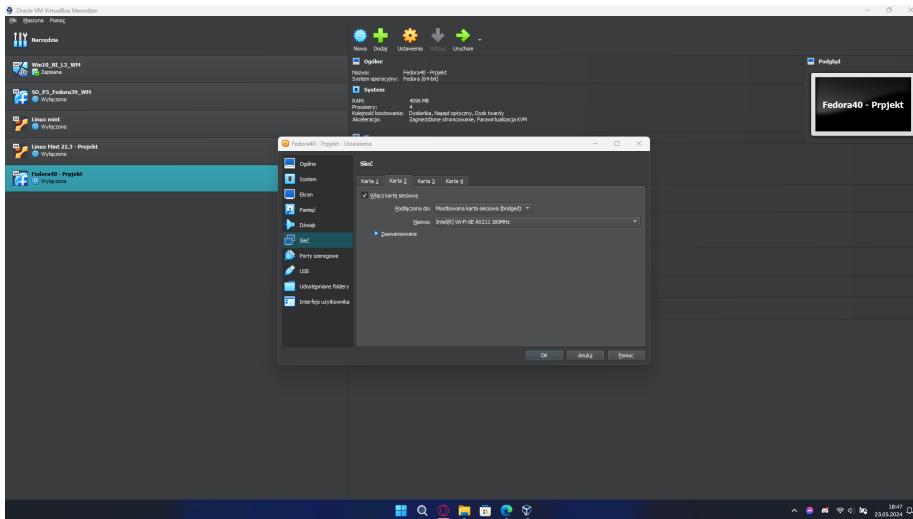
Rysunek 12: Analogicznie jak w przypadku instalacji Linux Mint – wymagane jest ustawienie nazwy maszyny wirtualnej, przydzielenie jej zasobów, ustalenie rozmiaru dysku. Powyższe zdjęcie ukazuje ekran z podsumowaniem wybranych opcji

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



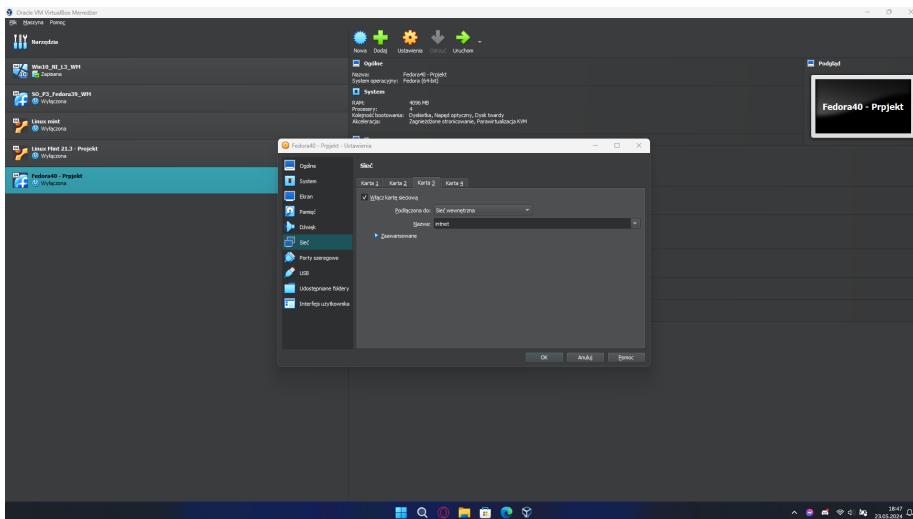
Rysunek 13: Dodanie pierwszej karty sieciowej – NAT

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.



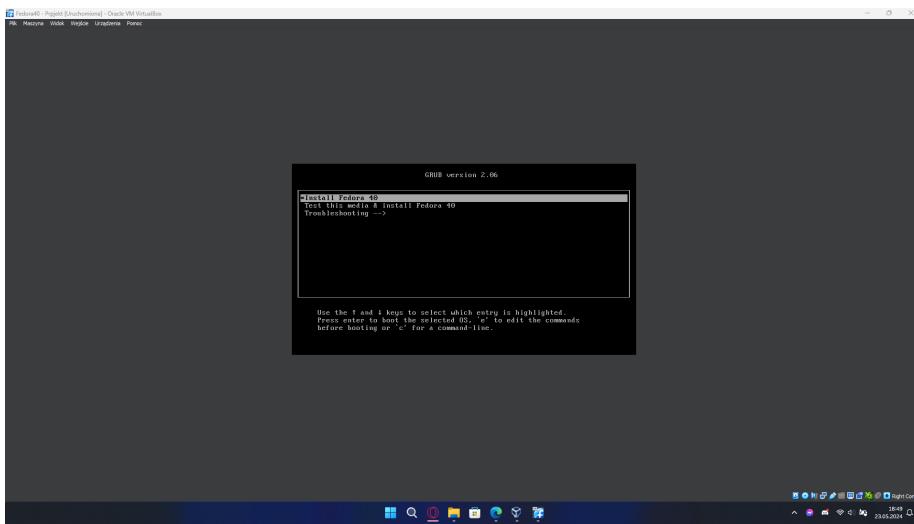
Rysunek 14: Dodanie drugiej karty sieciowej – sieć mostkowana (bridged)

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.



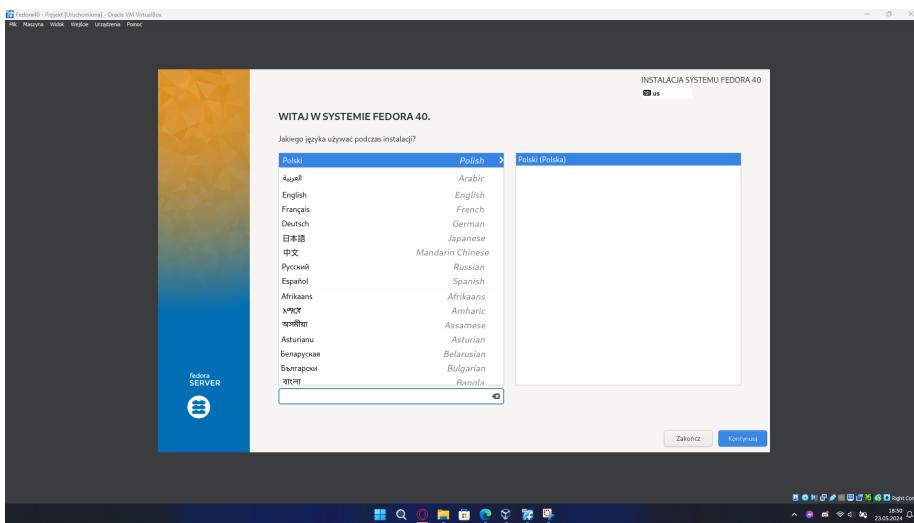
Rysunek 15: Dodanie trzeciej karty sieciowej – sieć wewnętrzna

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.



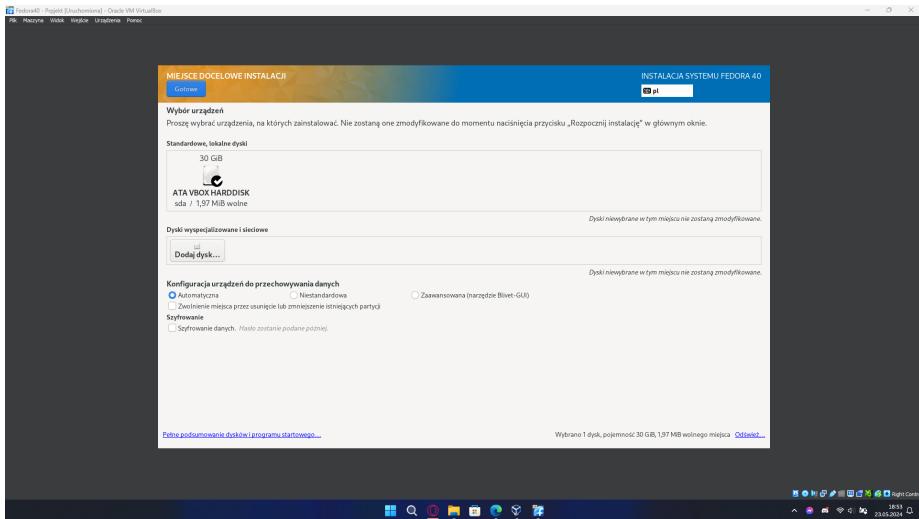
Rysunek 16: Uruchomienie instalatora Fedory.

W kolejnym kroku wybiera się język instalatora oraz układ klawiatury.



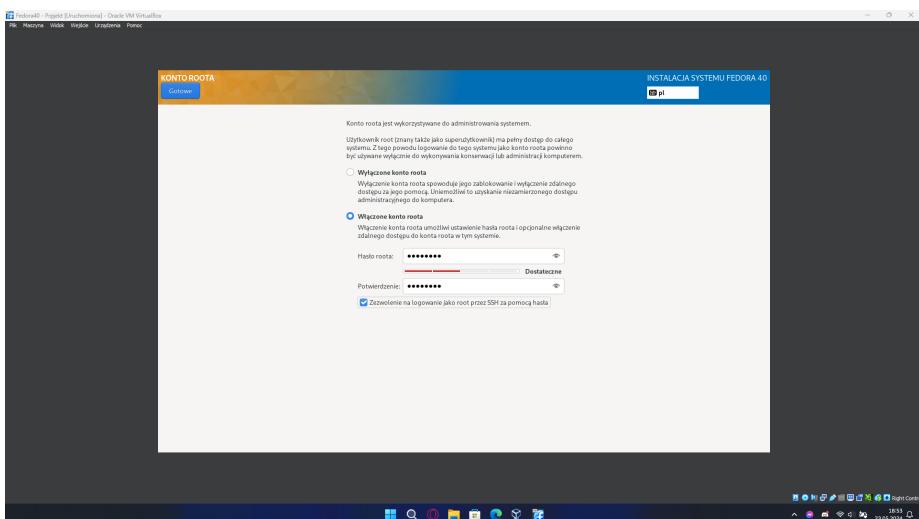
Rysunek 17: Rozpoczęcie instalacji Fedora 40 – wybór języka instalacji.

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



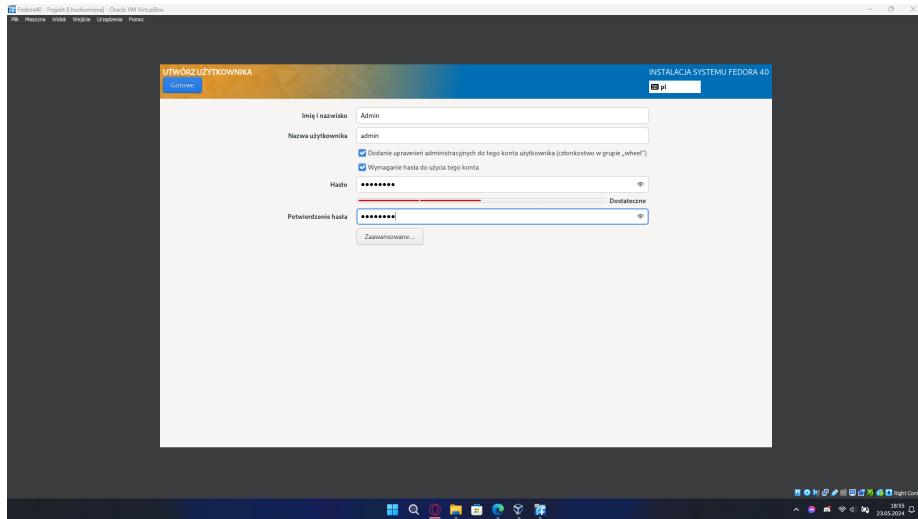
Rysunek 18: Wybór dysku na którym zostanie zainstalowany serwer

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



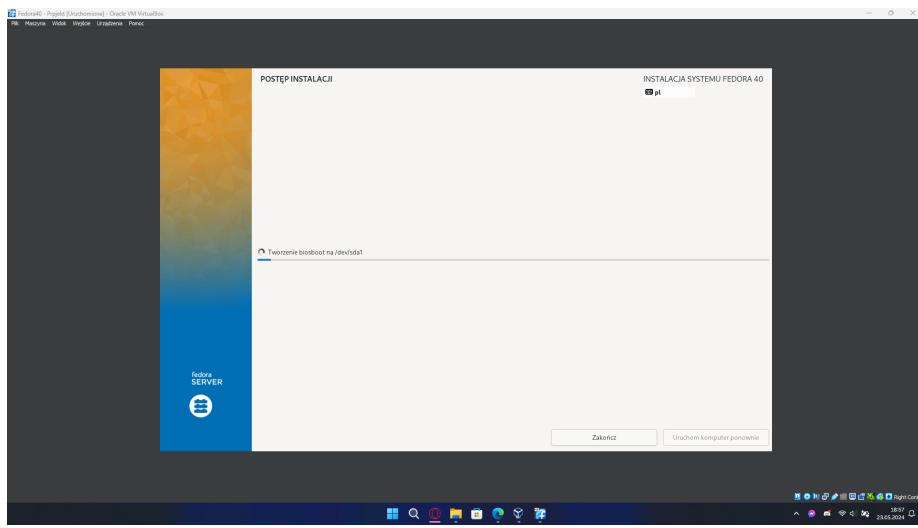
Rysunek 19: Ustawienie konta root – włączenie konta, ustawienie hasła i zezwolenie na połączenie ssh jako root

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



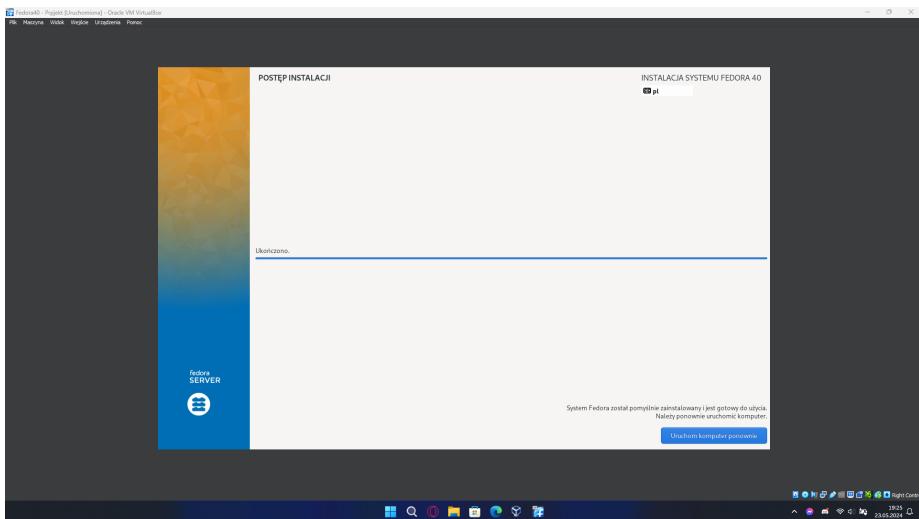
Rysunek 20: Stworzenie użytkownika – admin

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



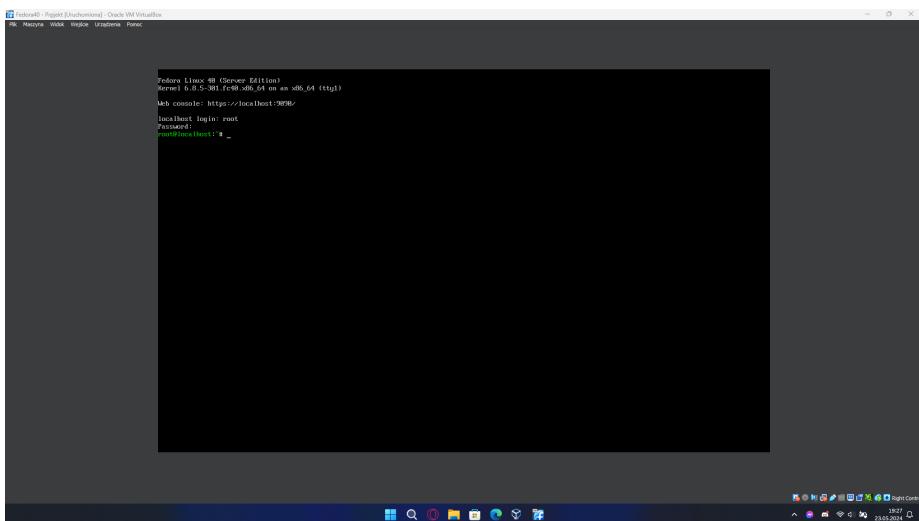
Rysunek 21: Ekran postępującej instalacji

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



Rysunek 22: Ekran postępującej instalacji – koniec instalacji

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



Rysunek 23: Zainstalowany system – przed wstępnią konfiguracją

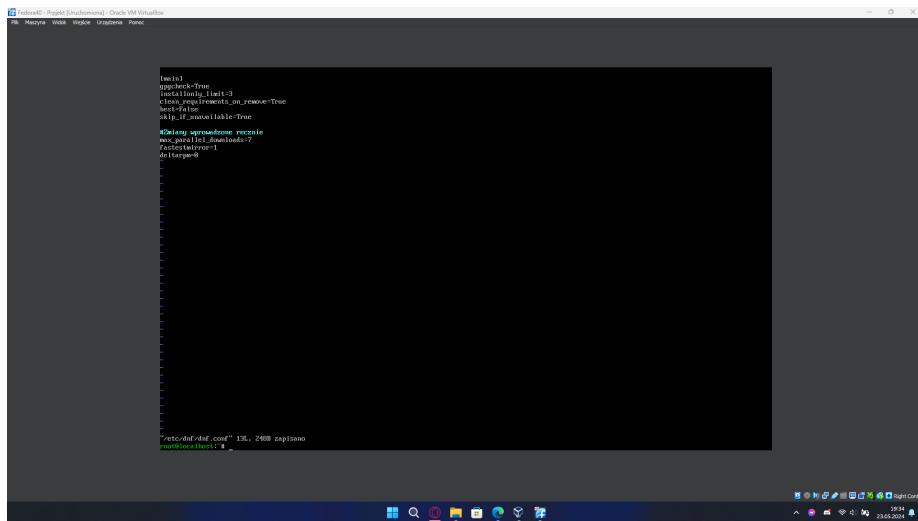
#### 4.2.2 Wstępna konfiguracja

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

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

Wytłumaczenie opcji:

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



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

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

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

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

Teraz po skonfigurowaniu menadżera pakietów można wykonać aktualizację pakietów.

Rysunek 25: Aktualizacja pakietów systemowych – test konfiguracji dnf

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

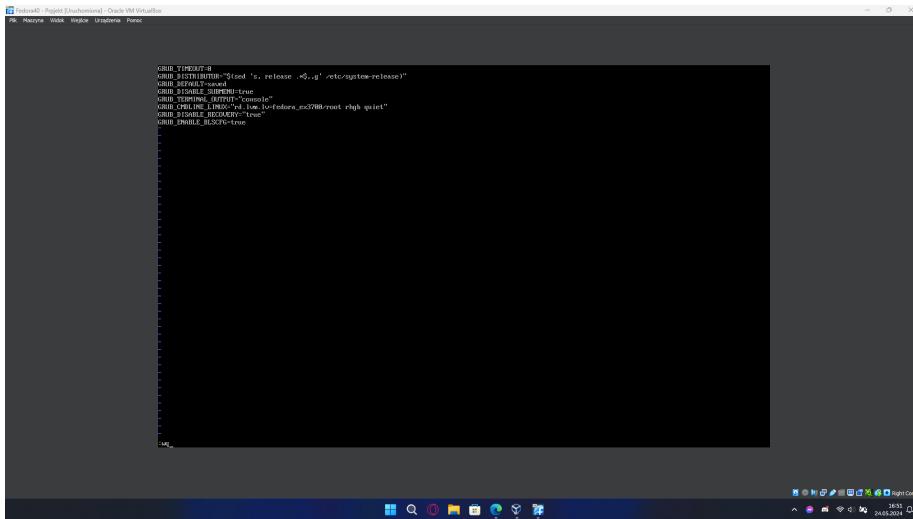
```
Fedora40 - Projekt [Unuchoming] - Oracle VM VirtualBox
```

```
GRUB_TIMEOUT=5
GRUB_BIOSITOR="$(sed 's,release .*,g' /etc/system-release)"
GRUB_DISABLE_LINUX_LDR=1
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL_OUTPUT="text"
GRUB_DISABLE_LINUX="true"
GRUB_DISABLE_EFI="true"
GRUB_ENABLE_RECOVERY="true"
GRUB_ENABLE_HIDDEN_MENU="true"

[...]
```

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

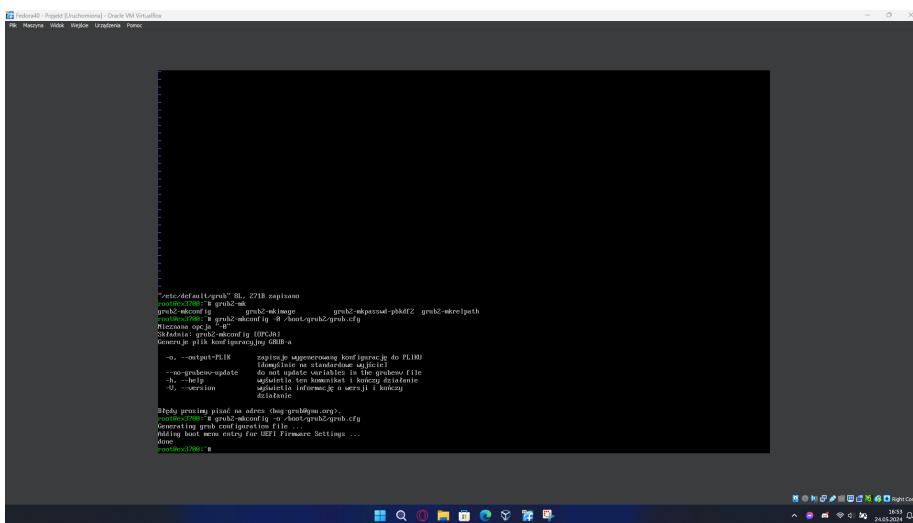
W kolejnym kroku zmieniłem GRUB\_TIMEOUT=5 na GRUB\_TIMEOUT=0  
Co można zauważyc poniższym zdjeciu.



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

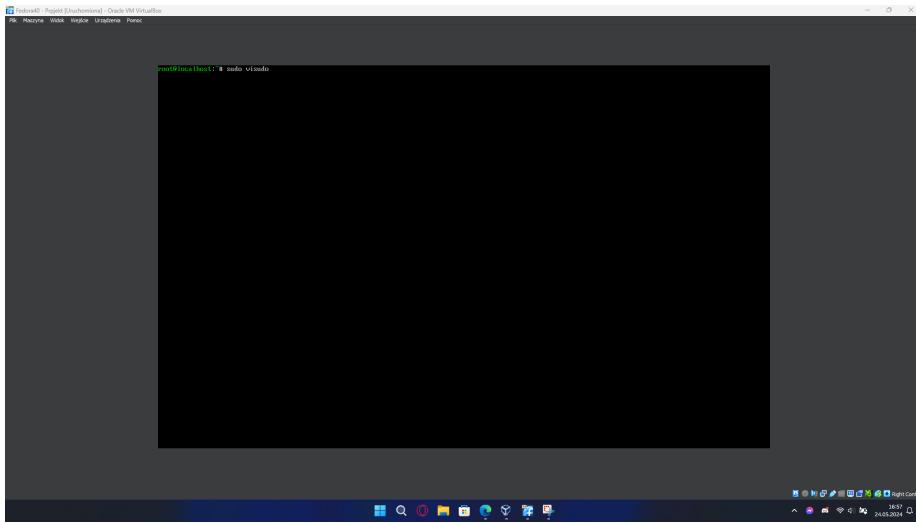
Aby zatwierdzić zmiany należy użyć komendy:

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



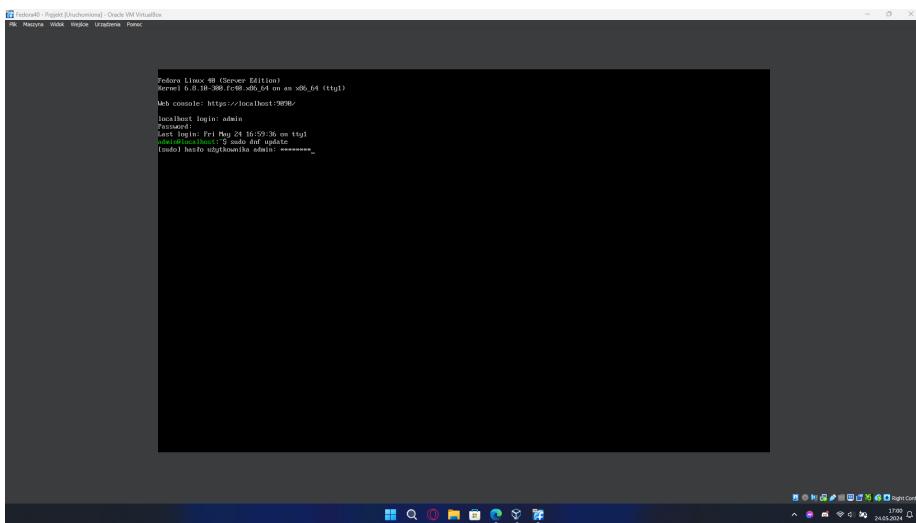
Rysunek 28: Zastosowanie zmian po edycji grub

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



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

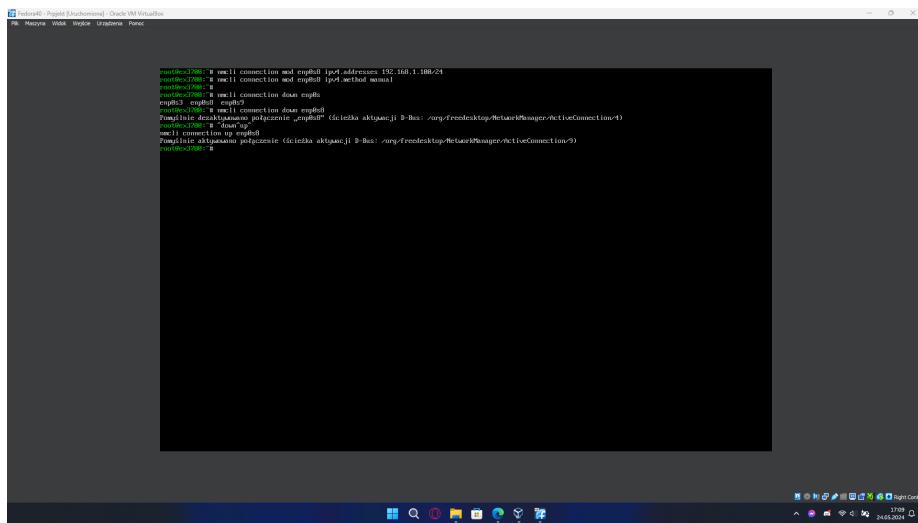
Efekt powyższego kroku:



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

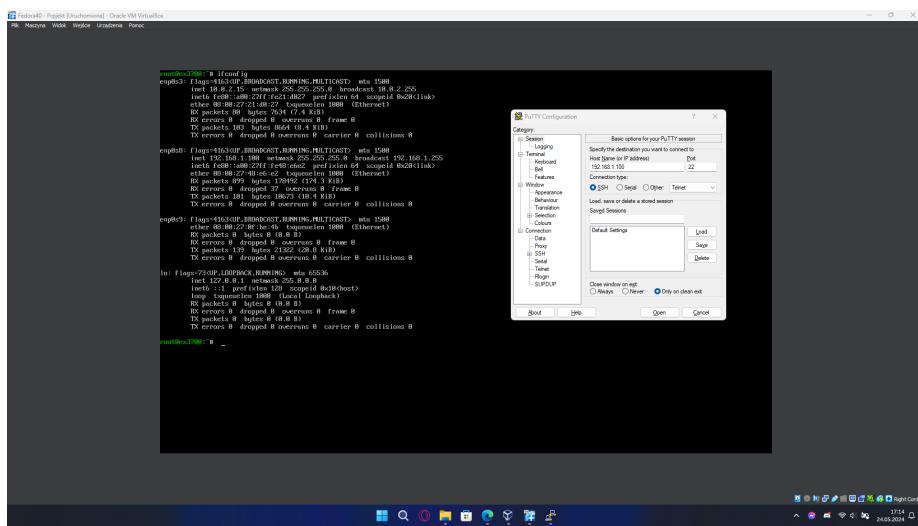
### 4.3 Konfiguracja SSH

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



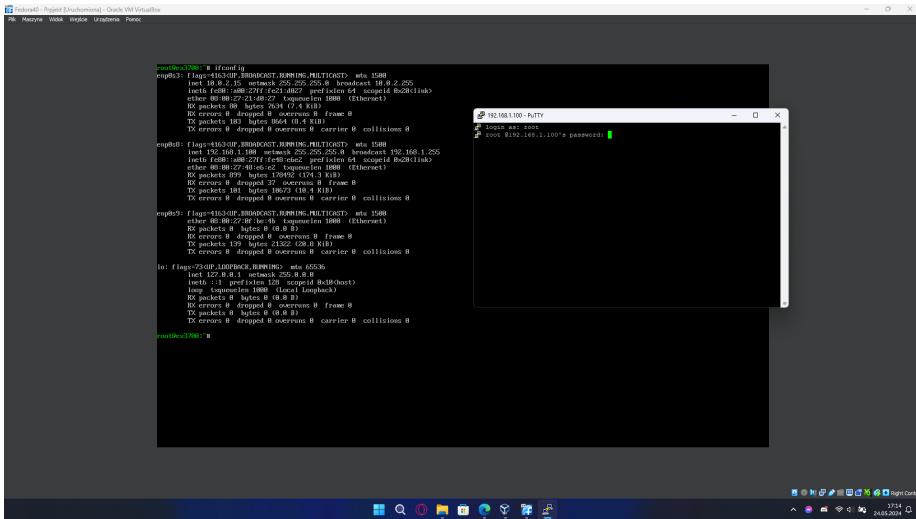
Rysunek 31: konfiguracja karty sieciowej

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



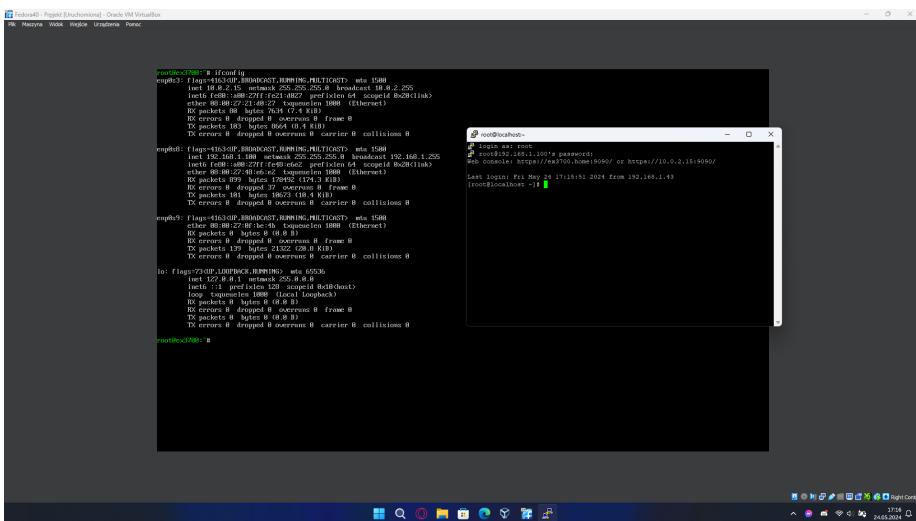
Rysunek 32: Konfiguracja aplikacji PuTTY

Próba zalogowania na konto root'a:



Rysunek 33: Podlaczenie poprzez PuTTY na konto root'a

Wynik powyższego kroku:

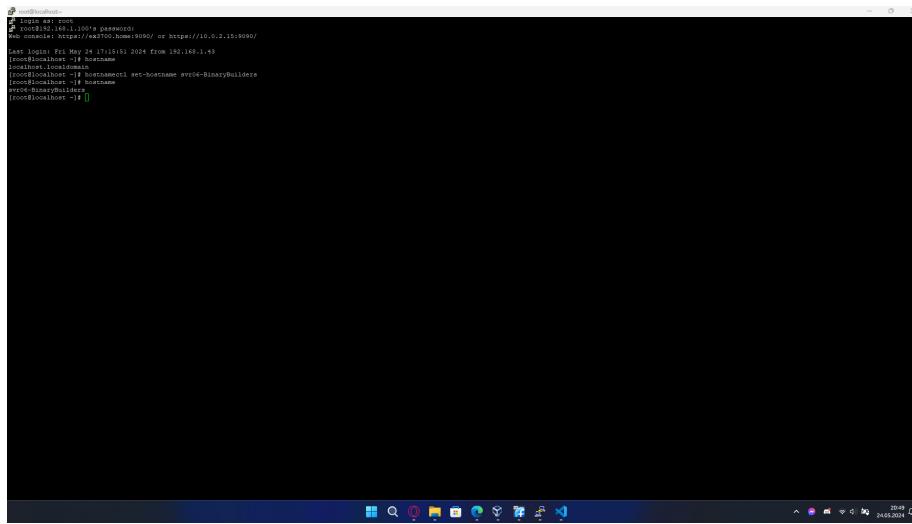


Rysunek 34: Wynik połączenia poprzez PuTTY

## 4.4 Nazwa serwera – hostname

Aby zmienić nazwę serwera (hostname) można użyć komendy:

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



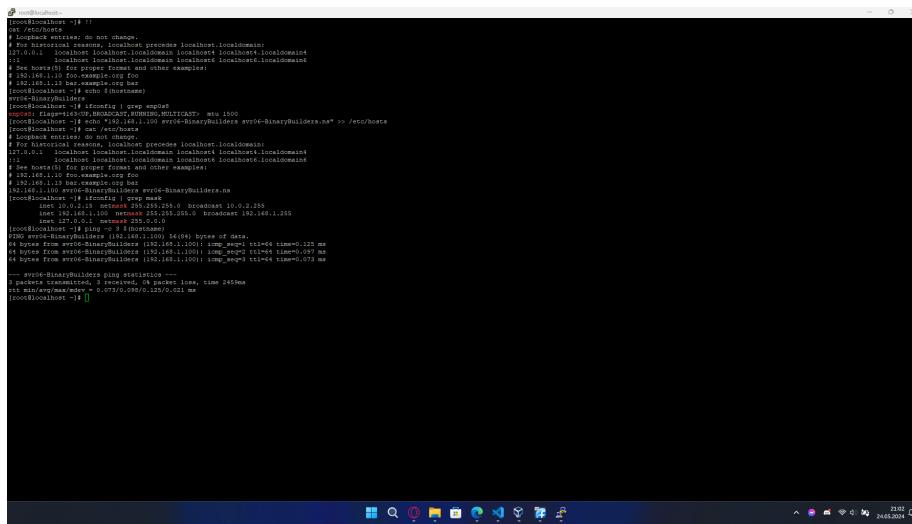
```
[root@svr04 ~]# hostnamectl set-hostname svr04-BinaryBuilders
[root@svr04 ~]#
```

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: flags=4163UP,BROADCAST,NOARP,MULTICAST mtu 1500
[root@svr04 ~]# ping -c 1 192.168.1.100 svr06-BinaryBuilders
PING svr06-BinaryBuilders (192.168.1.100) 56(84) bytes of data:
64 bytes from svr06-BinaryBuilders (192.168.1.100): icmp_seq=1 ttl=64 time=0.125 ms
64 bytes from svr06-BinaryBuilders (192.168.1.100): icmp_seq=2 ttl=64 time=0.097 ms
64 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 246ms
rtt min/avg/max/mdev = 0.073/0.096/0.125/0.021 ms
[root@svr04 ~]#
```

Rysunek 36: Edycja /etc/hosts

Aby zainstalować oprogramowanie do stworzenia serwera DNS należy wydać polecenie:

- Jeśli jesteś na koncie root:

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

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

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

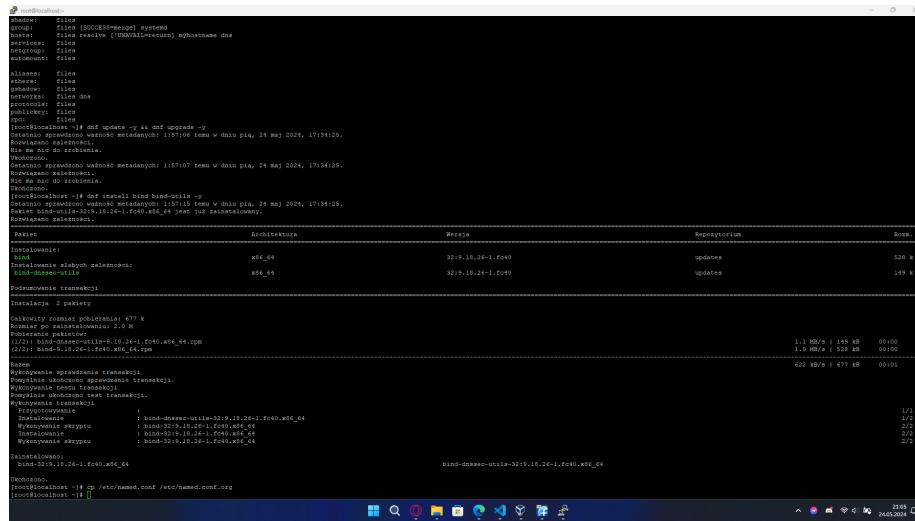
Po zainstalowaniu wymaganych pakietów należy wykonać kopię zapasową plików konfiguracyjnych. Można to zrobić komendą:

- Jeśli jesteś na koncie root:

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

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

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



Rysunek 37: Instalacja DNS

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

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

```

root@cockhost:~# cp /etc/named.conf /etc/named.conf.org
root@cockhost:~# du -h
4.0K    /etc/named.conf
4.0K    /etc/named.conf.org

```

Rysunek 38: Stworzenie kopii zapasowej pilku konfiguracyjnego DNS

```

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

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

```

```

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

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

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

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

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

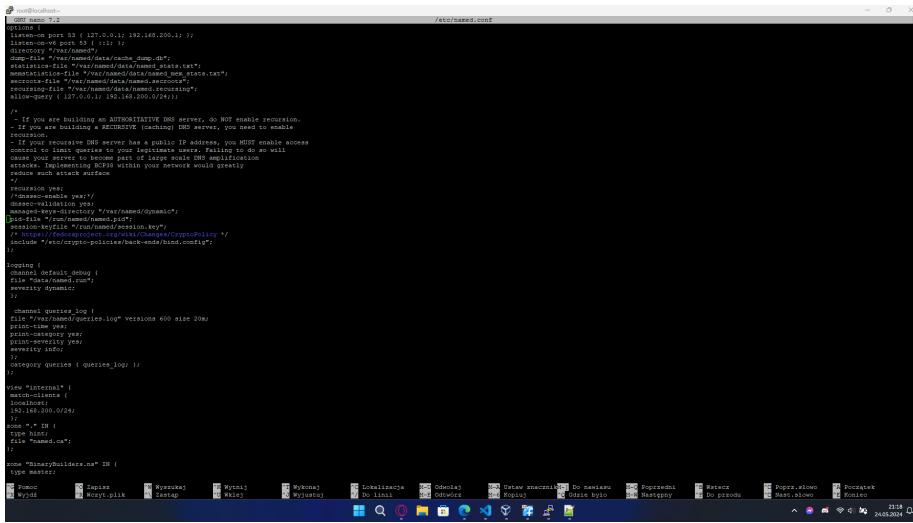
zone "BinaryBuilders.ns" IN {
    type master;
    file "BinaryBuilders.ns.lan_in";
    allow-update { none; };
};

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

```

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

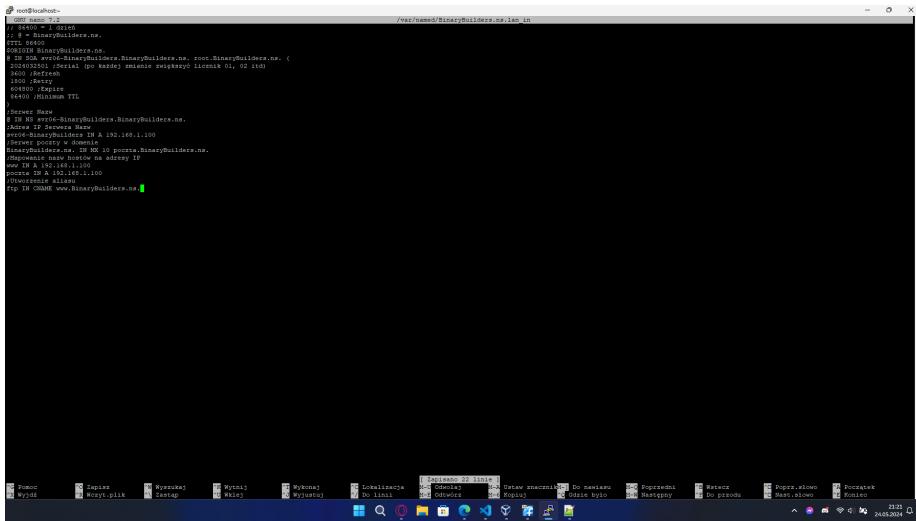
W kolejnym kroku trzeba utworzyć plik strefy podstawowej. W moim przypadku jest to plik /var/BinaryBuilders.ns.lan.in. Zawartość tego pliku:



Rysunek 39: zawartość named.conf

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

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



Rysunek 40: zawartość pliku strefy podstawowej

```

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

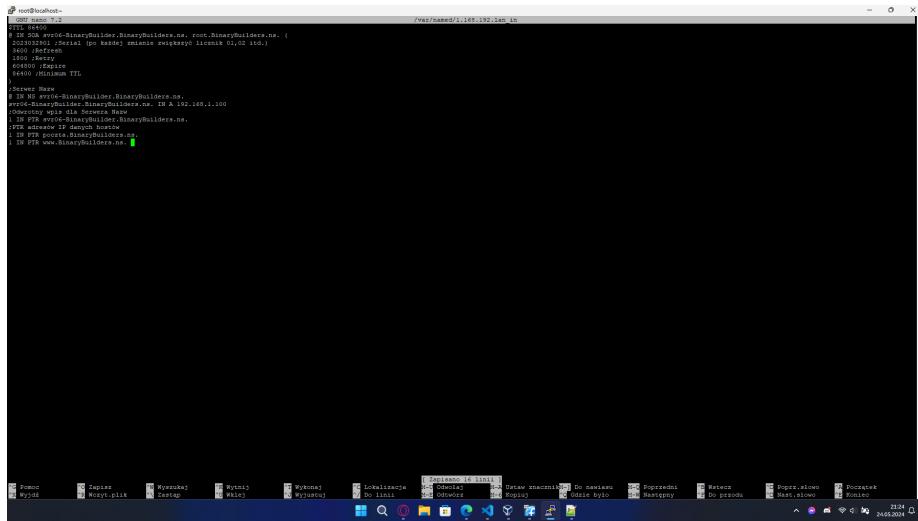
```

Następnym krokiem jest uruchomienie kilku komend:

```

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

```



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

Przeładowuje konfigurację serwera serwera DNS

- www.dz-studium

[View 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.

```
root@localhost ~
```

root@localhost ~]# /etc/init.d/named restart  
[named@localhost ~]#  
root@localhost ~]# http://list.susecon.org/mailman/listinfo/system-devel  
From: KacperSieradz [mailto:kacper.sieradz@wp.pl]  
Subject: Re: [system-devel] named: failed to start named.service - Berkeley Internet Name Domain (DNS).  
Date: Mon, 24 Oct 2016 15:58:27 +0200  
To: Kacper Sieradz [mailto:kacper.sieradz@wp.pl]  
Cc: susecon@list.susecon.org  
Message-ID: <20161024135827.10000-1-kacper.sieradz@wp.pl>  
Content-Type: text/plain; charset="utf-8"  
Content-Transfer-Encoding: 8bit  
User-Agent: Mozilla/5.0 (X11; Linux x86\_64; rv:45.0) Gecko/20100101 Thunderbird/45.0  
Content-Disposition: inline  
  
From: KacperSieradz [mailto:kacper.sieradz@wp.pl]  
Subject: Re: [system-devel] named: failed to start named.service - Berkeley Internet Name Domain (DNS).  
Date: Mon, 24 Oct 2016 15:58:27 +0200  
To: Kacper Sieradz [mailto:kacper.sieradz@wp.pl]  
Cc: susecon@list.susecon.org  
Message-ID: <20161024135827.10000-1-kacper.sieradz@wp.pl>  
Content-Type: text/plain; charset="utf-8"  
Content-Transfer-Encoding: 8bit  
User-Agent: Mozilla/5.0 (X11; Linux x86\_64; rv:45.0) Gecko/20100101 Thunderbird/45.0  
Content-Disposition: inline  
  
From: KacperSieradz [mailto:kacper.sieradz@wp.pl]  
Subject: Re: [system-devel] named: failed to start named.service - Berkeley Internet Name Domain (DNS).  
Date: Mon, 24 Oct 2016 15:58:27 +0200  
To: Kacper Sieradz [mailto:kacper.sieradz@wp.pl]  
Cc: susecon@list.susecon.org  
Message-ID: <20161024135827.10000-1-kacper.sieradz@wp.pl>  
Content-Type: text/plain; charset="utf-8"  
Content-Transfer-Encoding: 8bit  
User-Agent: Mozilla/5.0 (X11; Linux x86\_64; rv:45.0) Gecko/20100101 Thunderbird/45.0  
Content-Disposition: inline  
  
From: KacperSieradz [mailto:kacper.sieradz@wp.pl]  
Subject: Re: [system-devel] named: failed to start named.service - Berkeley Internet Name Domain (DNS).  
Date: Mon, 24 Oct 2016 15:58:27 +0200  
To: Kacper Sieradz [mailto:kacper.sieradz@wp.pl]  
Cc: susecon@list.susecon.org  
Message-ID: <20161024135827.10000-1-kacper.sieradz@wp.pl>  
Content-Type: text/plain; charset="utf-8"  
Content-Transfer-Encoding: 8bit  
User-Agent: Mozilla/5.0 (X11; Linux x86\_64; rv:45.0) Gecko/20100101 Thunderbird/45.0  
Content-Disposition: inline  
  
From: KacperSieradz [mailto:kacper.sieradz@wp.pl]  
Subject: Re: [system-devel] named: failed to start named.service - Berkeley Internet Name Domain (DNS).  
Date: Mon, 24 Oct 2016 15:58:27 +0200  
To: Kacper Sieradz [mailto:kacper.sieradz@wp.pl]  
Cc: susecon@list.susecon.org  
Message-ID: <20161024135827.10000-1-kacper.sieradz@wp.pl>  
Content-Type: text/plain; charset="utf-8"  
Content-Transfer-Encoding: 8bit  
User-Agent: Mozilla/5.0 (X11; Linux x86\_64; rv:45.0) Gecko/20100101 Thunderbird/45.0  
Content-Disposition: inline  
  
From: KacperSieradz [mailto:kacper.sieradz@wp.pl]  
Subject: Re: [system-devel] named: failed to start named.service - Berkeley Internet Name Domain (DNS).  
Date: Mon, 24 Oct 2016 15:58:27 +0200  
To: Kacper Sieradz [mailto:kacper.sieradz@wp.pl]  
Cc: susecon@list.susecon.org  
Message-ID: <20161024135827.10000-1-kacper.sieradz@wp.pl>

Rysunek 42: Uruchomienie usługi DNS

#### 4.6 DHCP – instalacja i konfiguracja

Aby zainstalować oprogramowanie do stworzenia serwera DHCP należy wydać polecenie:

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

Rysunek 43: Instalacja DHCP

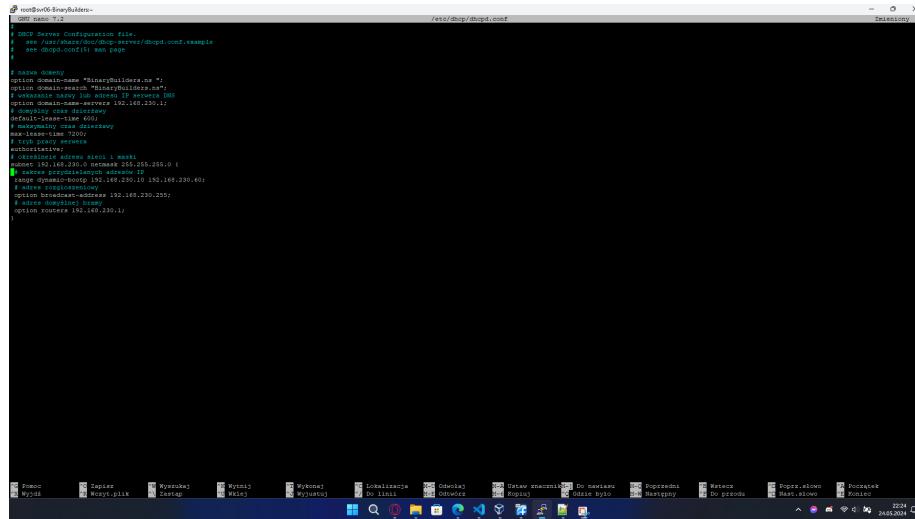
Po zainstalowaniu odpowiednich pakietów dobrze jest wykonać kopię zapasową oryginalnego pliku konfiguracji, co widać na zrucie powyżej. Można to zrobić następującą komendą:

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

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

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

Wprowadzoną treść widać na zrzucie poniżej.



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

Następnym krokiem jest wprowadzenie kilku poleceń:

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

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.

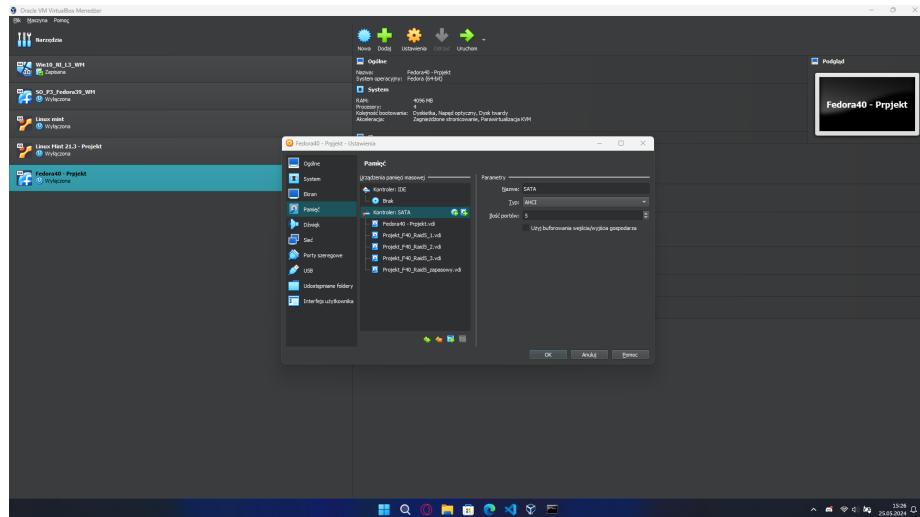
```
[root@srv06-BinaryBuilders ~]# nano /etc/dhcp/dhcpd.conf  
[root@srv06-BinaryBuilders ~]# systemctl start dhcpcd  
[root@srv06-BinaryBuilders ~]# systemctl enable dhcpcd  
[root@srv06-BinaryBuilders ~]# useradd -r -u 1000 -g 1000 -s /bin/false -c 'dhcpd' -m -d /var/lib/centos/dhcpd - /usr/lib/systemd/system/dhcpd.service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

```

[root@centos-vm ~]# mdadm --create /dev/md0 --level=5 --raid-devices=3 /dev/sd[b-d] --spare-devices=1 /dev/sde
mdadm: layout defaults to left-symmetric
mdadm: chunk size defaults to 64K
mdadm: array /dev/md0 started.
mdadm: /dev/sdb[0]:0 active sync 1.00G /dev/sdb[0]
mdadm: /dev/sdc[0]:0 active sync 1.00G /dev/sdc[0]
mdadm: /dev/sdd[0]:0 active sync 1.00G /dev/sdd[0]
mdadm: /dev/sde[0]:0 spare 1.00G /dev/sde[0]
mdadm: /dev/sde[0] is Superblock is persistent
  Update Time : Sat May 25 19:54:12 2008
    State : clean
  Active Devices : 3
 Working Devices : 4
 Failed Devices : 0
 Spare Devices : 1

  Persistence: Superblock is persistent

  Layout: left-symmetric
  Chunk Size: 64K

  Resync Policy: 100%_linear

  Array UUID: 7e4f7d2d:0e705977:2a11:e1bd1fbc
  Events: 18

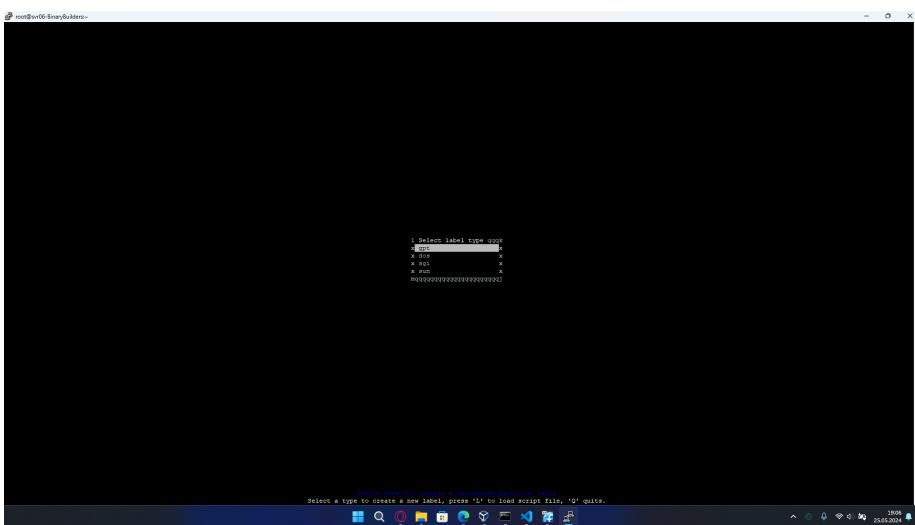
Number Major Minor RaidDevice State
  0   8      32       0      active sync  /dev/sdb
  1   8      32       1      active sync  /dev/sdc
  2   8      32       2      active sync  /dev/sdd
  3   8      44       3      spare none   /dev/sde

[root@centos-vm ~]#

```

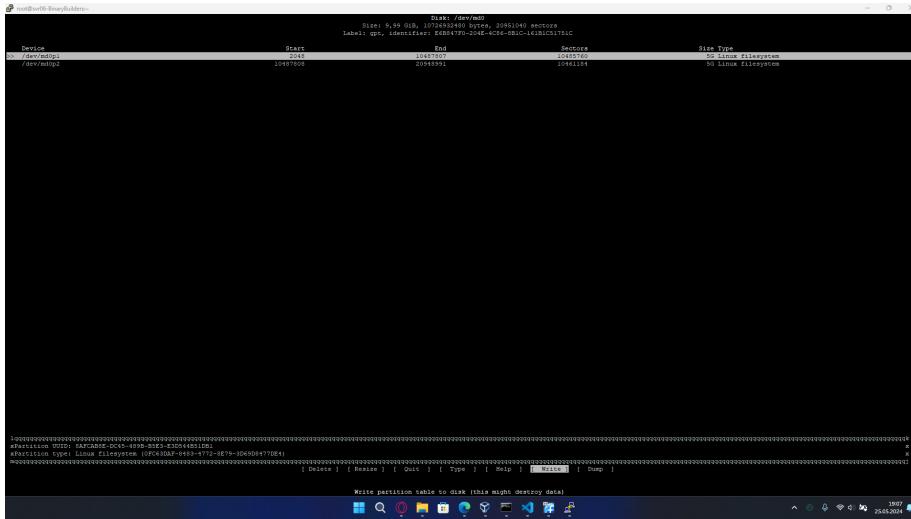
Rysunek 47: Stworzenie macierzy raid 5

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



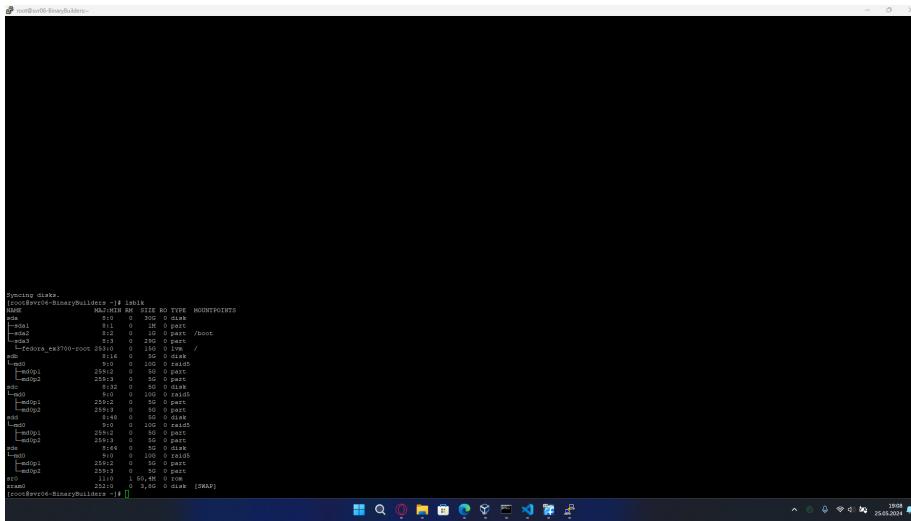
Rysunek 48: Partycjonowanie macierzy narzędziem cfdisk

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



Rysunek 49: Stworzenie dwóch partycji – każda 5GB

Potwierdzenie działania poprzedniej komendy:



Rysunek 50: Wynik partycjonowania

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

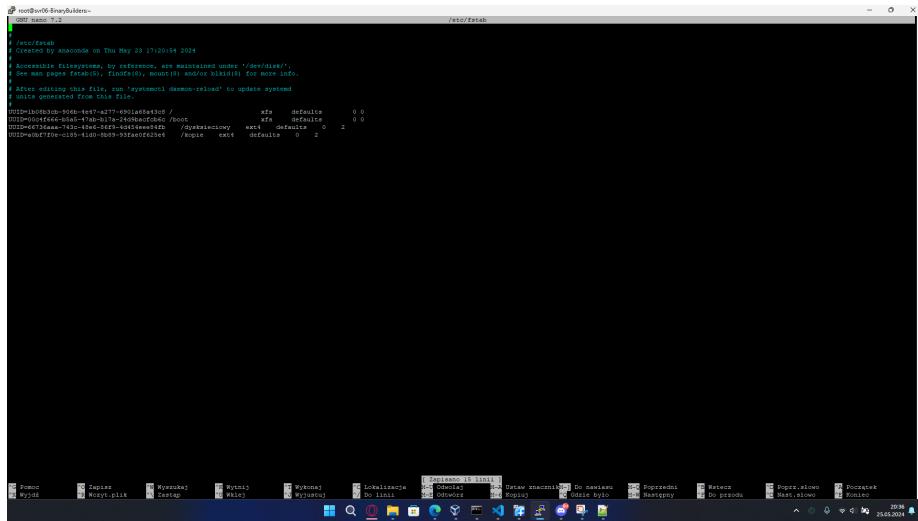
```
[root@Fedor-OptiPlex-5070 ~]# cd /mnt
[root@Fedor-OptiPlex-5070 mnt]# rm -rf *
[root@Fedor-OptiPlex-5070 mnt]# mkdir -p /dysksieciowy /kopie
[root@Fedor-OptiPlex-5070 mnt]# mount /dev/sda1 /dysksieciowy
[root@Fedor-OptiPlex-5070 mnt]# mount /dev/sda2 /kopie
[root@Fedor-OptiPlex-5070 mnt]# df -Th
Filesystem      Type   Size  Used Avail Mount point
/dev/sda1        ext4   100G  100G   0B  /dysksieciowy
/dev/sda2        ext4   100G  100G   0B  /kopie
[root@Fedor-OptiPlex-5070 ~]#
```

Rysunek 51: Przygotowanie ścieżek do montowania

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

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

UUID=1b08b3cb-906b-4e47-a277-6901a68a43c8 /
    ↳ xfs    defaults      0 0
UUID=00c4f666-b5a5-47ab-b17a-24d9bacfcbb6c /boot
    ↳ xfs    defaults      0 0
UUID=66736aaa-743c-48e6-86f9-4d454eee84fb    /dysksieciowy
    ↳ ext4  defaults      0  2
UUID=a0bf7f0e-c185-41d0-8b89-93fae0f625e4    /kopie    ext4
    ↳ defaults 0  2
```



Rysunek 52: Edycja /etc/fstab

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

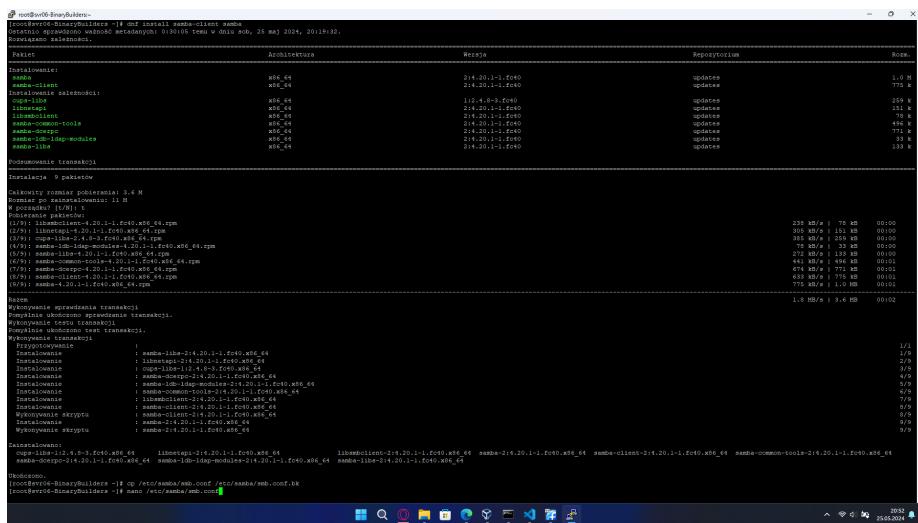
## 4.8 Samba – instalacja i konfiguracja

Aby zainstalować oprogramowanie do stworzenia serwera samba należy wydać polecenie:

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

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

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



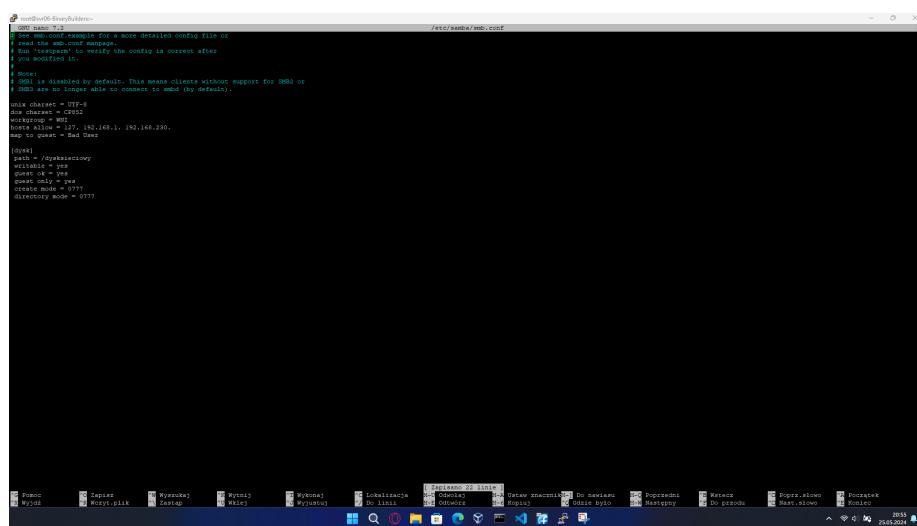
Rysunek 53: Samba – instalacja

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

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

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

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



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

Następnym krokiem jest wprowadzenie kilku polecień:

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

Wytłumaczenie powyższych polecień:

- testparm

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

- systemctl start smb nmb

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

- systemctl enable smb nmb

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

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

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

- firewall-cmd --reload

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

- setsebool -P samba\_export\_all\_rw on

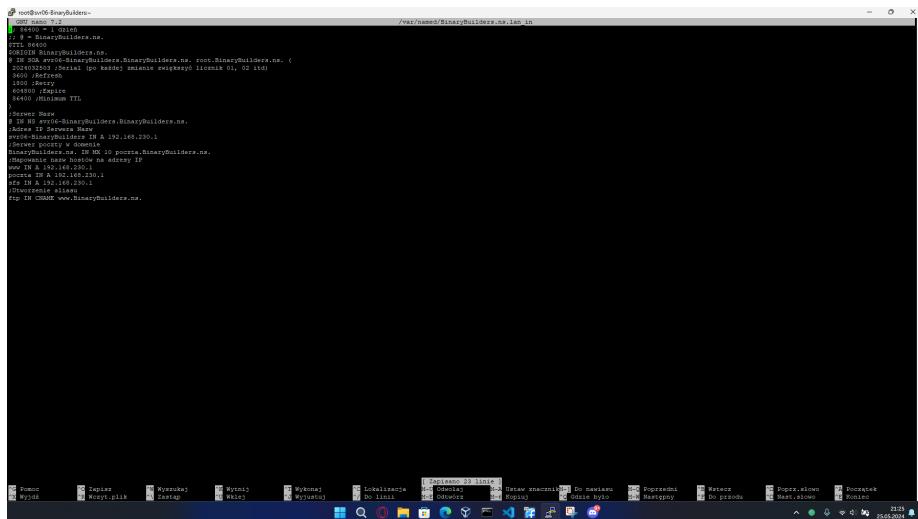
Ustawia w SELinux politykę, która pozwala Sambie na eksportowanie wszystkich udziałów z prawami do odczytu i zapisu. Dzięki temu Samba może zarządzać plikami z pełnym dostępem zgodnie z ustawieniami SELinux.

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

Rysunek 55: Samba – ustawienia SELinux oraz firewall

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

sfs IN A 192.168.230.1



Rysunek 56: Edycja konfiguracji DNS

Po 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
```

```
[root@localhost ~]# dnf install httpd -y
[...]
[root@localhost ~]#
```

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

```
[root@localhost ~]# cat /etc/httpd/conf/httpd.conf
[...]
[root@localhost ~]#
```

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>

```

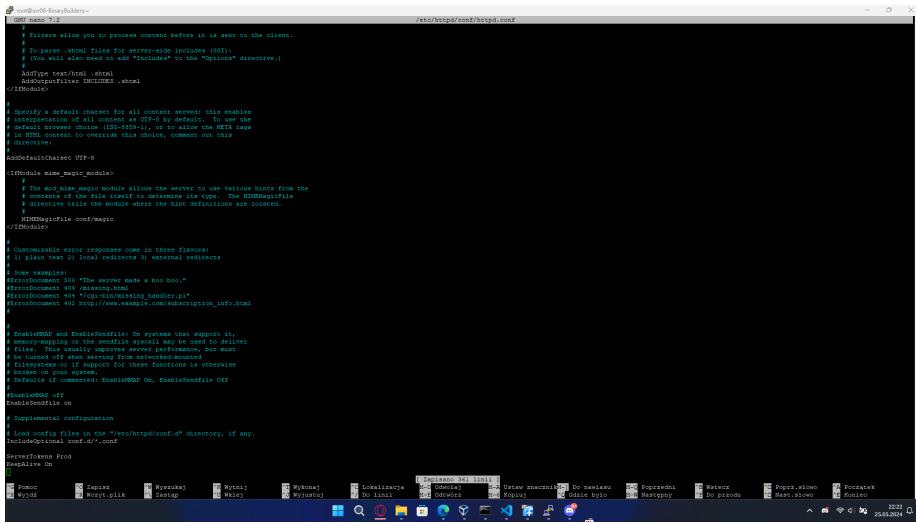
Rysunek 59: Edycja /etc/httpd/conf/httpd.conf – część druga

```

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



```
[root@www05 ~]# more /etc/httpd/conf/httpd.conf
#
# Filters allow you to process content before it is sent to the client.
# 
# (Note: if you are using SSI, then you will need to uncomment this)
# (You will also need to add "Includes" to the "Options" directive.)
# AddType text/html .shtml
AddOutputFilter INCLUDES .shtml
VIMHOME

# Specify a default charset for all content served; this enables
# your server to handle international characters correctly.
# Default browser choice (ISO-8859-1), or to allow the META tag
# (which is the preferred way to override this choice, comment out this
# directive).
AddDefaultCharset UTF-8

<IfModule mod_mime_module>
# The mod_mime module allows the server to use various hints from the
# contents of the file itself to determine its type. The XSendFile module
# requires this module to determine where the files are located.
# AddType application/x-javascript .js
# AddType application/x-shockwave-flash .swf
</IfModule>

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

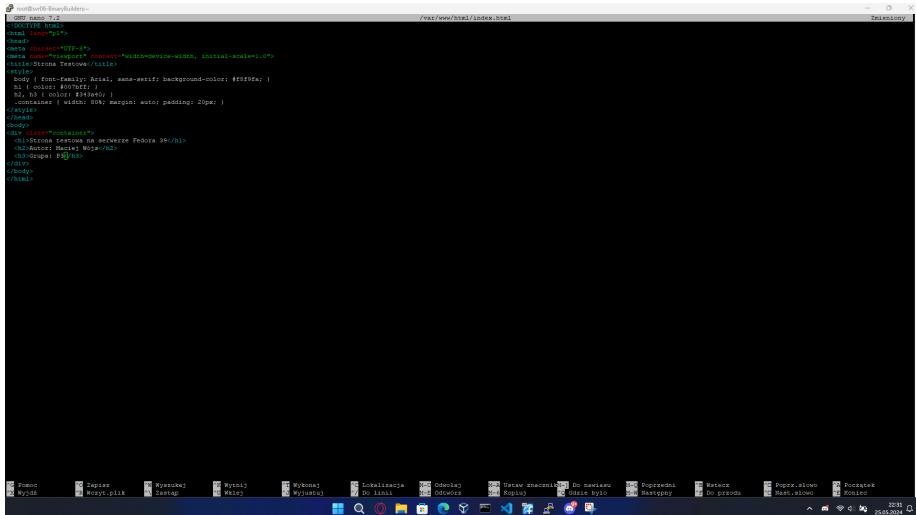
# 
# mod_deflate and mod_expires directives. On systems that support it,
# mod_expires in the mod_deflate module may be used to serve
# files. This usually improves server performance, but must
# be used with care as it can interfere with other
# filters or if support for these functions is otherwise
# disabled. If commented: EnableDEFLATE On, EnableSendfile Off
#EnableDEFLATE On
#EnableSendfile On
# Supplemental configuration
# Load config files in the /etc/httpd/conf.d/* directory, if any.
#IncludeOptional conf.d/*
<IfModule mod_expires.c>
ExpiresActive On
# Expire on
</IfModule>
```

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

W następnym kroku wykonuje następujące polecenia:

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

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



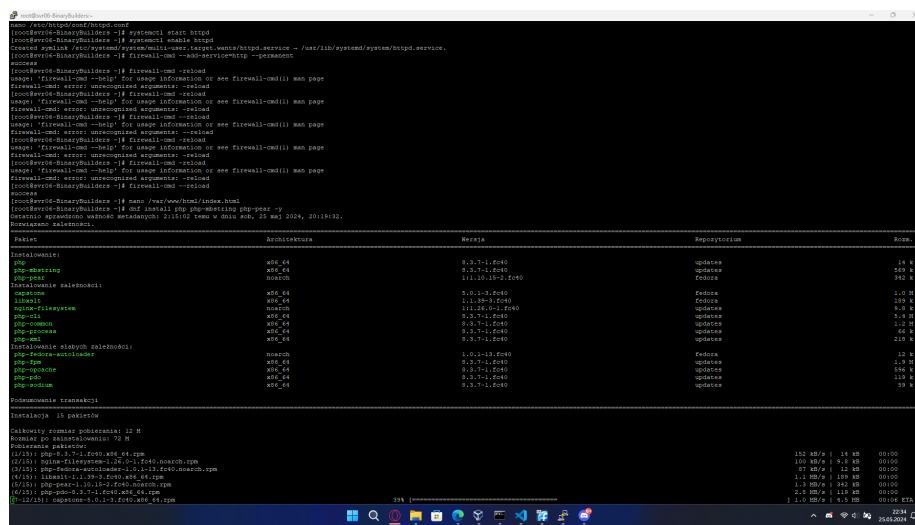
Rysunek 62: Strona html – domyślna strona serwera

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

## 4.10 PHP – instalacja i konfiguracja

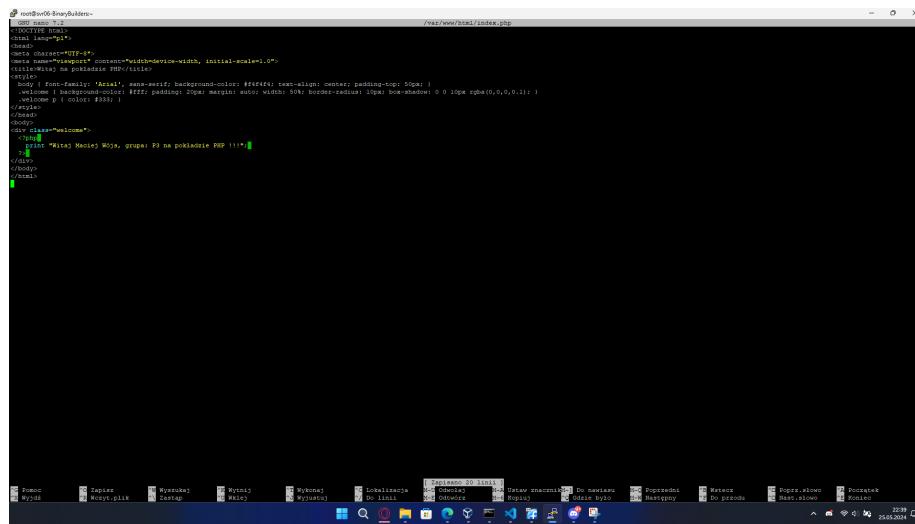
Aby zainstalować PHP należy wydać polecenie:

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



Rysunek 63: PHP – instalacja

Po zainstalowaniu wymaganych pakietów tworzę prostą stronę internetową wykorzystującą PHP.



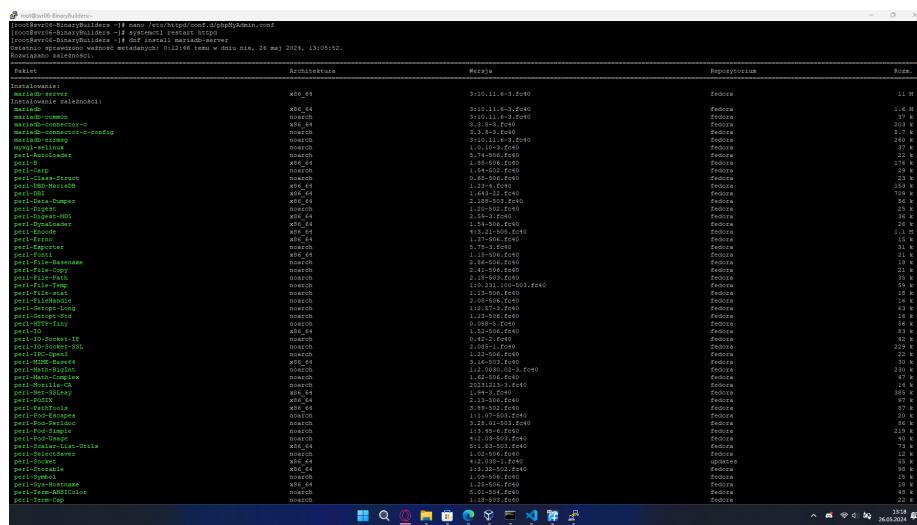
Rysunek 64: PHP – stworzenie strony internetowej

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

## 4.11 mariadb – instalacja i konfiguracja

Aby zainstalować silnik bazydanych mariadb należy wydać polecenie:

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



Rysunek 65: mariadb – instalacja usługi

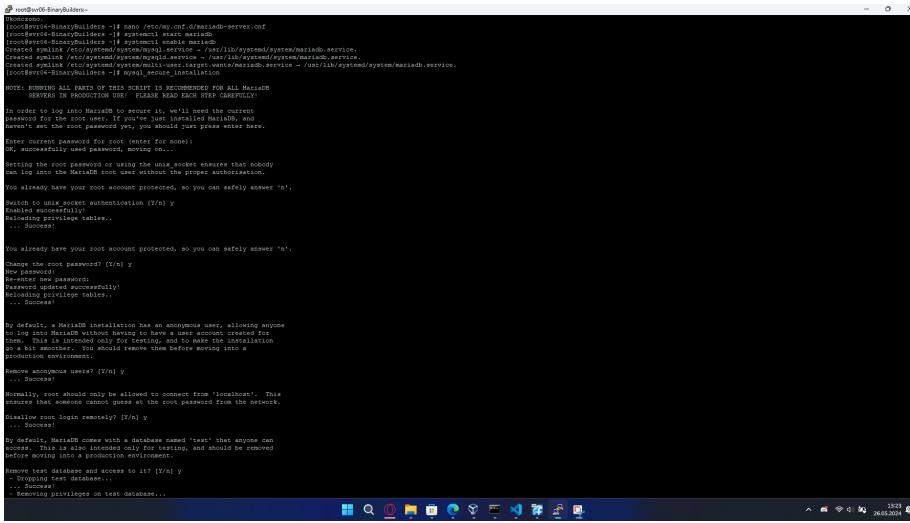
Do pliku /etc/my.cnf.d/mariadb-server.cnf w sekcji [mysqld] dodałem linię:

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



Rysunek 66: mariadb – edycja pliku konfiguracyjnego

Restart usługi mariadb oraz instalacja serwera MySQL.



```
root@DevW8-Server:~# ./mysql_secure_installation
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 can run MariaDB or MySQL with the '-u' option and enter the
password for the root user. If you've just installed MariaDB, and
haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):
+-----+
|          |
|          |
|          |
+-----+
Enter password:
Re-enter new password:
Password updated 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!

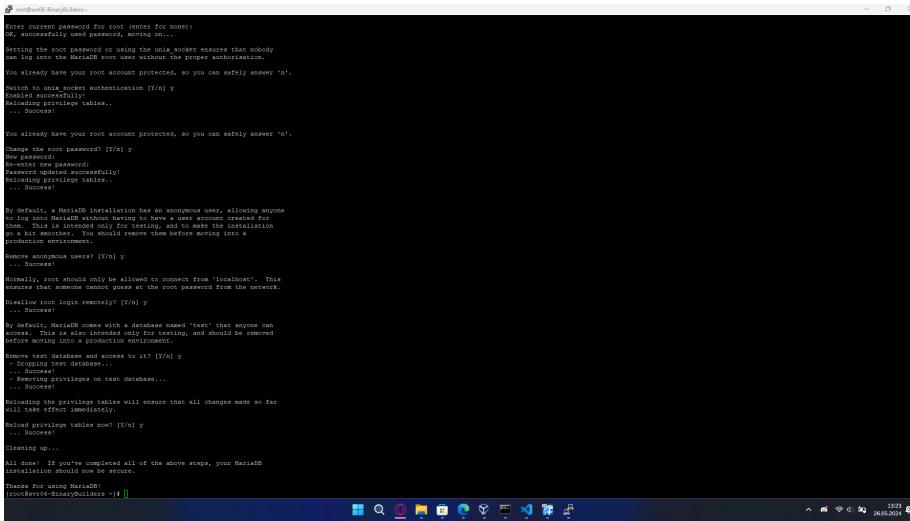
All privilege changes will ensure that all changes made so far
will take effect immediately.

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!
prospective-binary-builders -i |
```

Rysunek 67: MySQL – instalacja część pierwsza



```
root@DevW8-Server:~# ./mysql_secure_installation
Enter current password for root (enter for none):
+-----+
|          |
|          |
|          |
+-----+
Enter password:
Re-enter new password:
Password updated 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!
prospective-binary-builders -i |
```

Rysunek 68: MySQL – instalacja część druga

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

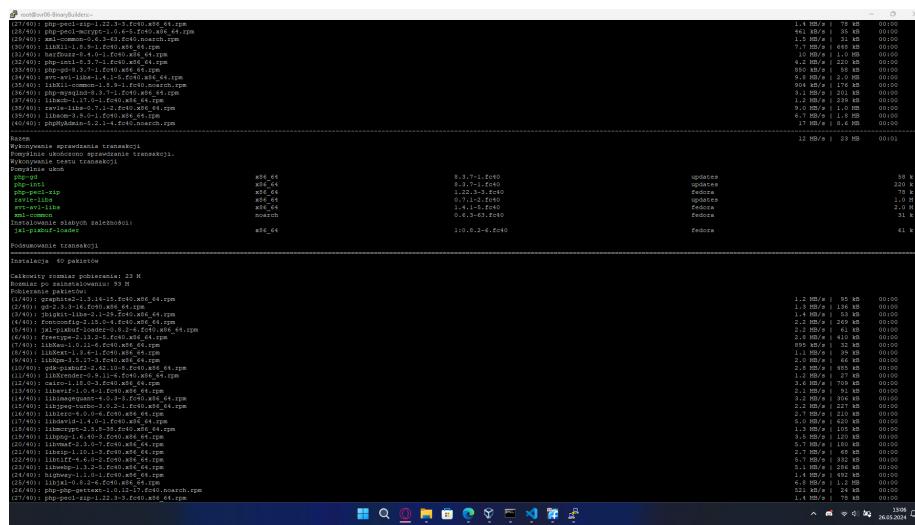
## 4.12 phpMyAdmin – instalacja i konfiguracja

Aby zainstalować phpMyAdmin należy użyć komendy:

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

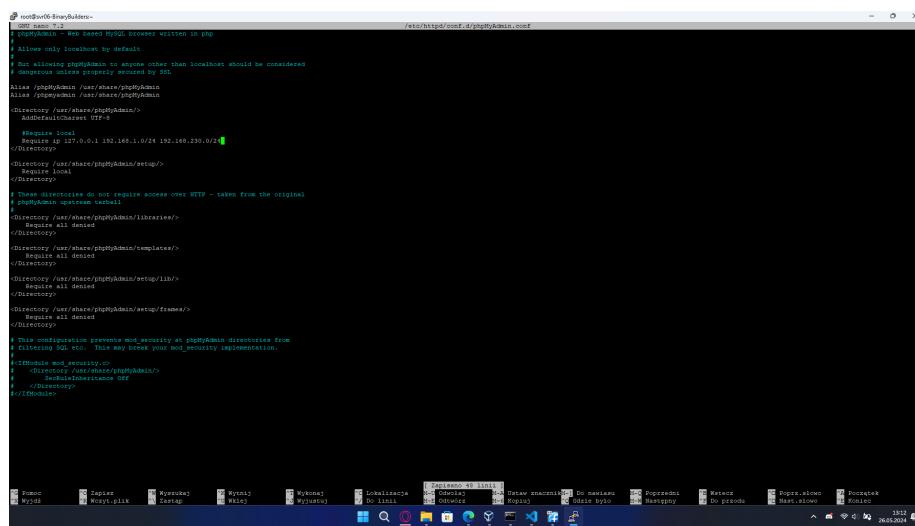
Po instalacji wykonuje kopię oryginalnego pliku konfiguracyjnego. Można to zrobić komendą:

```
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

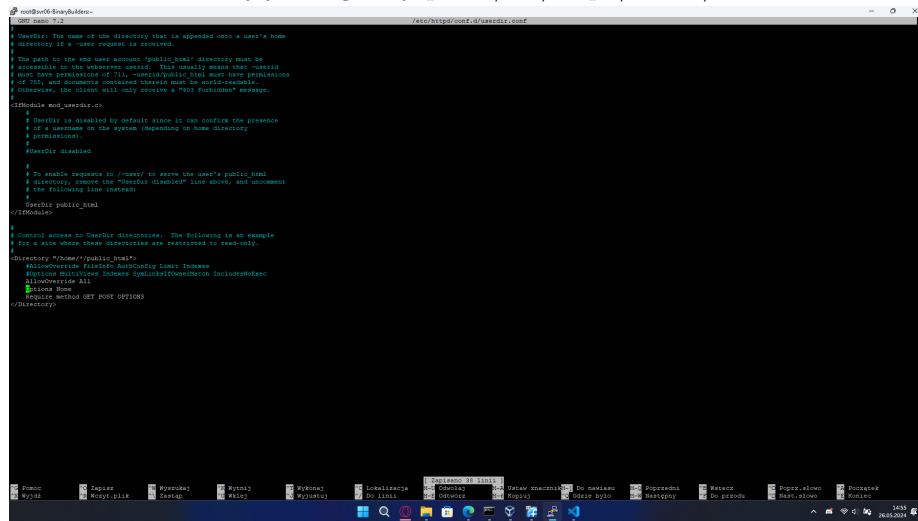
Test działania phpMyAdmin dostępny jest [tutaj](#).

## 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`



```
# UserDir: The name of the directory that is appended onto a user's home
# directory of a user's request is reversed.
#
# If the UserDir directive is present, the user's home directory must be
# accessible to the webserver userid. This usually means that 'userdir
# or 'rwx' permissions must be granted on the home directory or its
# parent directories. Note that the 'UserDir' directive must be world-readable
# otherwise, the client will only receive a "403 Forbidden" message.


    # UserDir is disabled by default since it can confuse the presence
    # of a user on the system (depending on home directory
    # permissions).
    #UserDir disabled

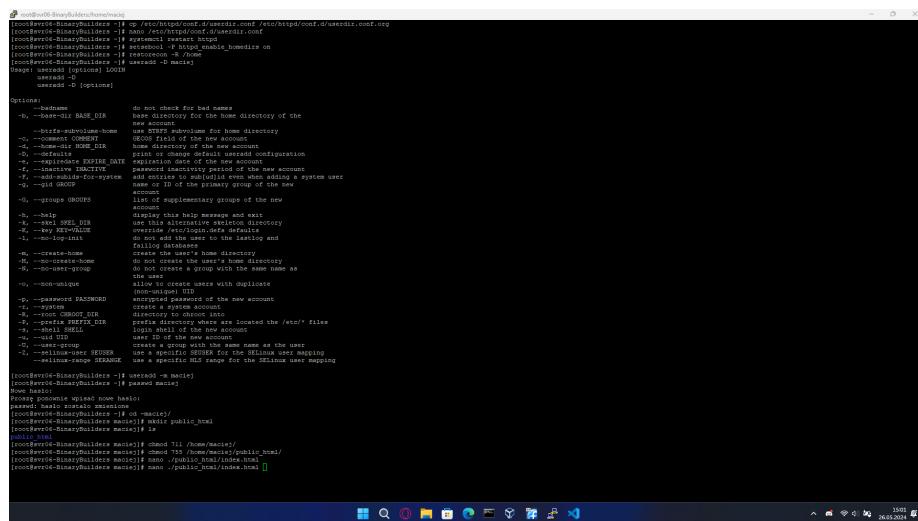
    #
    # To enable requests to /UserDir/ to serve the user's public_html
    # directory, add the following line instead:
    #
    # UserDir public_html
    #/ifModule

    Control access to UserDir directories. The following is an example
    for a site where these directories are restricted to read-only.
    Directory "/UserDir/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
    

```

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.

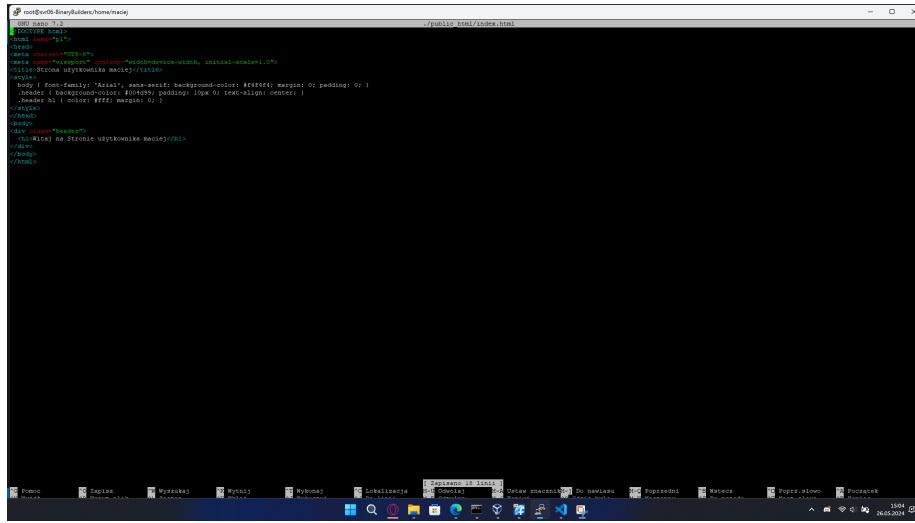


```
[root@vps04-BinaryBuilders ~]# op /etc/httpd/conf.d/userdir.conf /etc/httpd/conf.d/userdir.conf.org
[root@vps04-BinaryBuilders ~]# systemctl restart httpd
[root@vps04-BinaryBuilders ~]# restorecon -R /home
[root@vps04-BinaryBuilders ~]# semanage user -a -t user_t -c www_user_t maciej
maciej: useradd [options] LOGIN
        -u, --uid UID           user ID
        -g, --gid GID           group ID
        -o, --object-rules-for-system SYSTEM          object rules for system
        -s, --shell SHELL        default shell
        -c, --comment COMMENT   comment
        -m, --create-home       create the user's home directory
        -n, --no-create-home    do not create the user's home directory
        -p, --password PASSWORD user password
        -r, --non-unique         allow to create users with duplicate
                                (non-unique) GIDs
        -h, --help               print this help message and exit
        -s, --selinux SELINUX    use this alternative selinux directory
        -t, --tftpboot TFTPDIR   use this alternative tftpboot directory
        -l, --no-log-init        do not add the user to the loginit and
                                for initd, the /etc/init.d files
        -m, --create-home       create the user's home directory
        -n, --no-create-home    do not create the user's home directory
        -p, --password PASSWORD user password
        -r, --non-unique         allow to create users with duplicate
                                (non-unique) GIDs
        -s, --system             create a system account
        -t, --tftpboot TFTPDIR   prefix directory where are located the /etc/* files
        -p, --prefix PREFIXDIR   prefix directory where are located the /etc/* files
        -u, --uid UID            user ID of the new account
        -g, --gid GID            group ID of the new account
        -c, --comment COMMENT   create a group with the same name as the user
        -r, --realname REALNAME  real name of the user
        -s, --selinux-range SELANGE use a specific SELinux range mapping
        -e, --selinux-range SELANGE use a specific SELinux range for the SELinux user mapping

[root@vps04-BinaryBuilders ~]# useradd -t -c www_user_t maciej
maciej: useradd: cannot create home directory for maciej
New shell:
passwd: Maciej musi zmienić hasło.
passwd: hasło zostało zmienione.
[root@vps04-BinaryBuilders ~]# su - maciej
[root@vps04-BinaryBuilders maciej]# cd /home/maciej
[root@vps04-BinaryBuilders maciej]# mkdir public_html
[root@vps04-BinaryBuilders maciej]# ls
[root@vps04-BinaryBuilders maciej]# touch test.txt
[root@vps04-BinaryBuilders maciej]# chmod 755 /home/maciej
[root@vps04-BinaryBuilders maciej]# nano ./public_html/index.html
[root@vps04-BinaryBuilders maciej]# nano ./public_html/index.html
```

Rysunek 72: Dodanie użytkownika maciej

Kod html tej strony



```
[root@ev08-BinaryBuilders:home]# useradd -m maciej
[root@ev08-BinaryBuilders:home]# echo "maciej:maciej" | chpasswd
[maciej@ev08-BinaryBuilders ~]$ cat >index.html <>
<!DOCTYPE HTML>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>
<h1>Witaj na stronie użytkownika maciej!</h1>
</body>
</html>
```

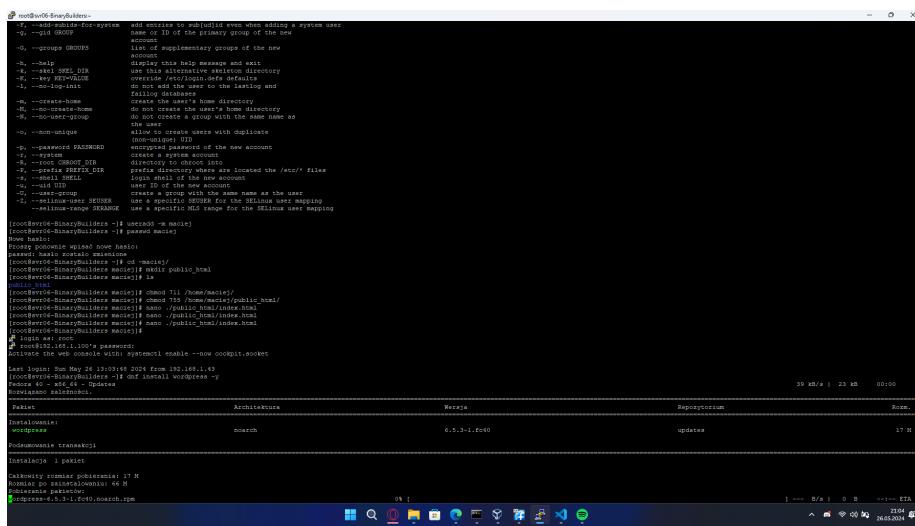
Rysunek 73: Stworzenie strony użytkownika maciej

Test działania dostępny jest [tutaj](#).

#### 4.14 WordPress – instalacja i konfiguracja

Aby zainstalować WordPress'a należy wykonać polecenie:

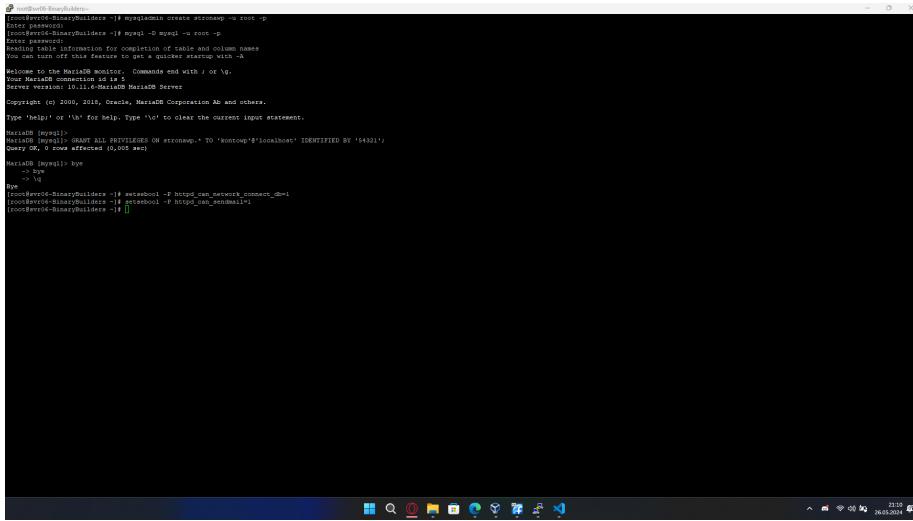
```
sudo dnf install wordpress -y
```



```
[root@ev08-BinaryBuilders ~]# useradd -m maciej
[maciej@ev08-BinaryBuilders ~]$ cd /var/www/public_html
[maciej@ev08-BinaryBuilders ~]$ nano index.html
[maciej@ev08-BinaryBuilders ~]$ ls
[maciej@ev08-BinaryBuilders ~]$ wp core install --url="http://192.168.1.100/maciej" --title="My Site" --admin_user="maciej" --admin_password="maciej" --admin_email="maciej@maciej.pl"
[maciej@ev08-BinaryBuilders ~]$ wp config update
[maciej@ev08-BinaryBuilders ~]$
```

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

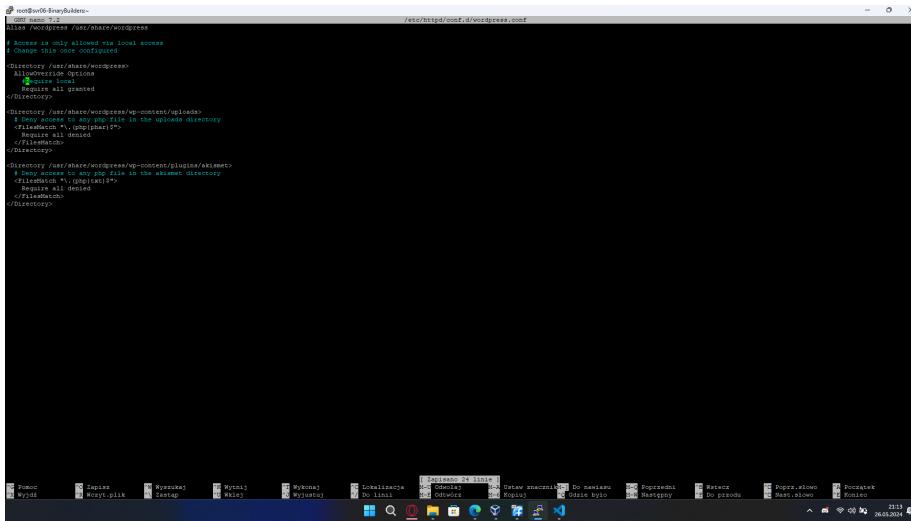
```
Alias /wordpress /usr/share/wordpress

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

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

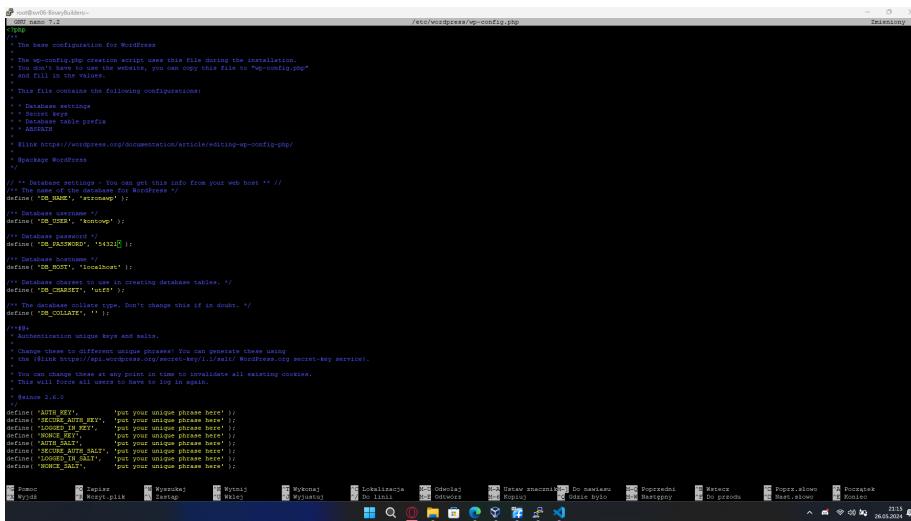
<Directory /usr/share/wordpress/wp-content/uploads>
    # Deny access to any php file in the uploads directory
    <FilesMatch "\.(php|phar)$">
        Require all denied
    </FilesMatch>
</Directory>

<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.

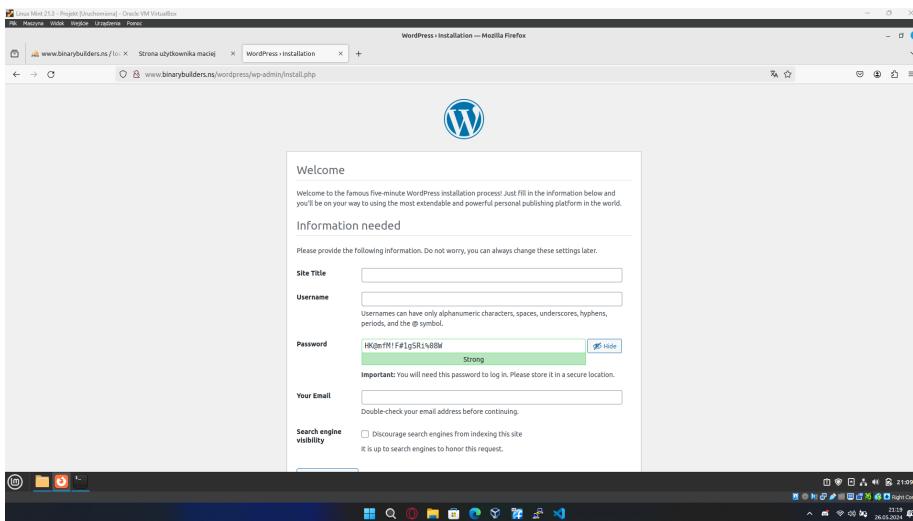


Rysunek 77: Edycja pliku /etc/wordpress/wp-config.php

W kolejnym kroku należy wykonać następujące polecenia:

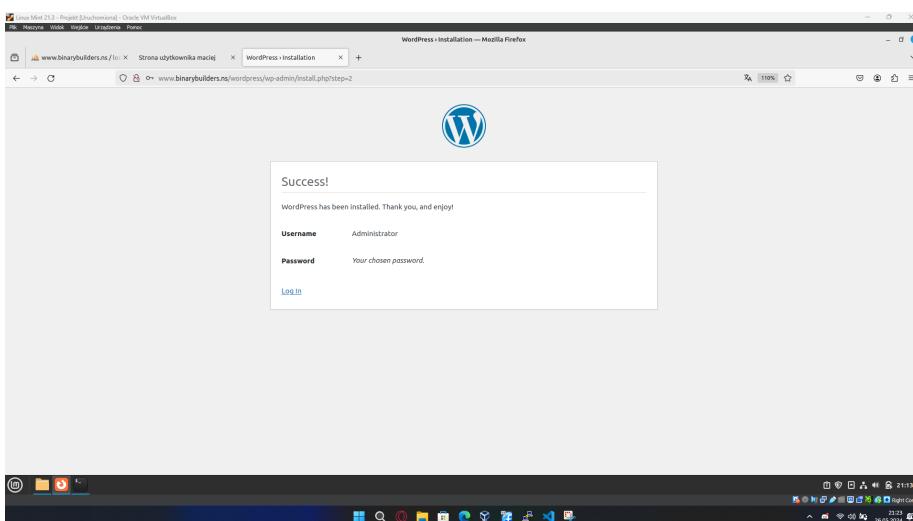
```
sudo setsebool -P httpd_can_network_connect_db=1
sudo setsebool -P httpd_can_sendmail=1
sudo systemctl restart httpd
```

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



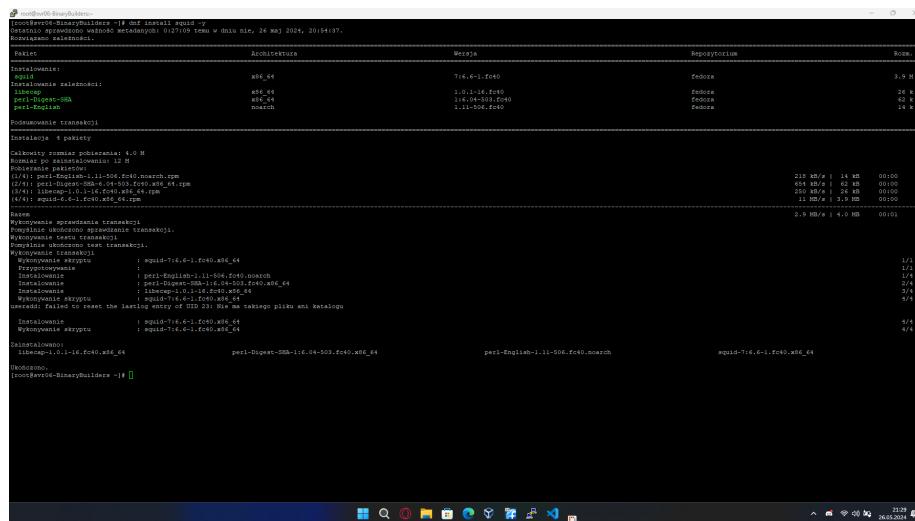
Rysunek 79: Instalacja WordPress – sukces

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

## 4.15 Proxy – instalacja i konfiguracja

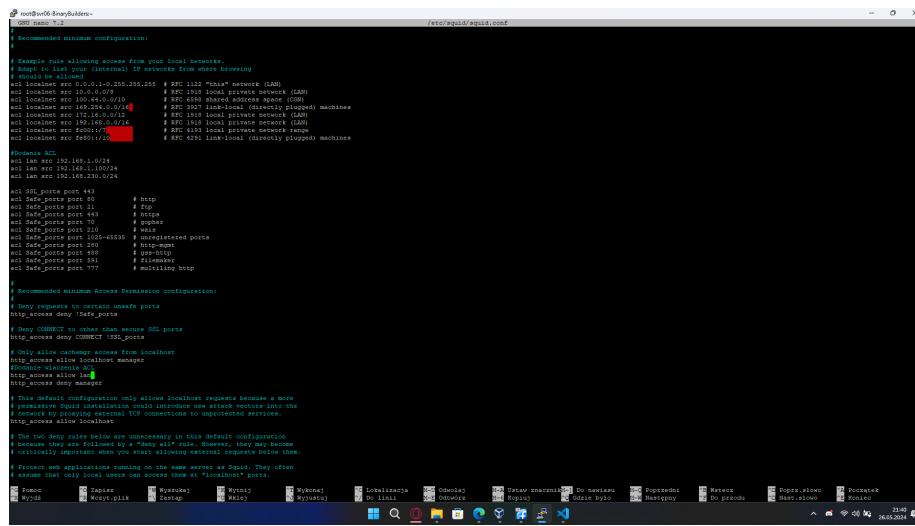
Aby zainstalować serwer proxy należy wydać polecenie:

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



Rysunek 80: Proxy – instalacja

W kolejnym kroku trzeba dodać do pliku /etc/squid/squid.conf zawartość jak w dwóch następujących 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](#).

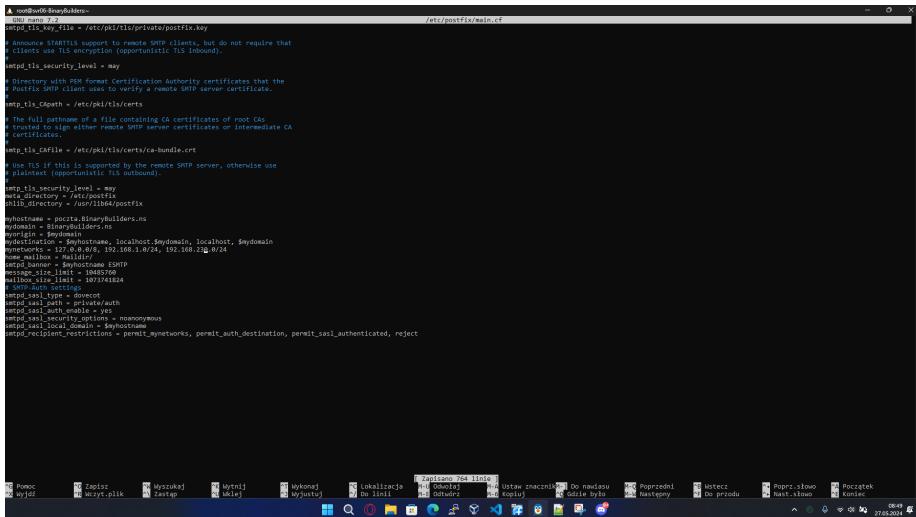
#### 4.16 SMTP – instalacja i konfiguracja

Komenda do instalacji SMTP:

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

Rysunek 83: Proxy – konfiguracja część druga

W następnym kroku trzeba dodać do pliku `/etc/postfix/main.cf` zawartość tak jak na dwóch następnych zdjęciach.

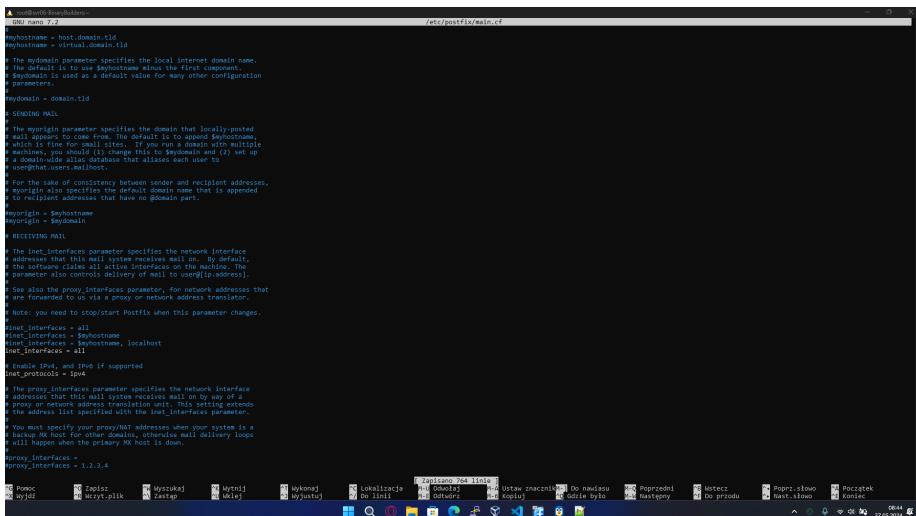


```

# mail postfix main.cf
Gmail name 7.2                                     /etc/postfix/main.cf
smtpd_tls_cert_file = /etc/pki/tls/private/postfix.key
smtpd_tls_key_file = /etc/pki/tls/private/postfix.key
# Announce STARTTLS support to remote SMTP clients, but do not require that
# clients use TLS encrypted (opportunistically TLS is allowed).
# smtpd_tls_security_level = may
# smtpd_tls_CAfile specifies a verification Authority certificates that the
# Postfix SMTP client uses to verify a remote SMTP server certificate.
# smtpd_tls_CACert = /etc/pki/tls/certs/ca-bundle.crt
# If TLS is not supported by the remote SMTP server, otherwise use
# plain text (opportunistically TLS is allowed).
# smtpd_tls_security_level = may
# smtpd_directory = /etc/postfix
# smtpd_recipient_restrictions = permit_mynetworks, reject
# myhostname = posta.BianyBuilders.ms
# mydomain = BianyBuilders.ms
# mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain
# home_mailbox = Maildir
# smtpd_banner = $myhostname ESMTP $mail_name ($mail_version)
# mailbox_size_limit = 5000000000
# mailbox_infinity_limit = 0
# smtpd_sasl_type = dovecot
# smtpd_sasl_auth_enable = yes
# smtpd_sasl_authenticated = yes
# smtpd_sasl_local_domain = $myhostname
# smtpd_recipient_restrictions = permit_mynetworks, permit_auth_destination, permit_sasl_authenticated, reject

```

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



```

# mail postfix main.cf
Gmail name 7.2                                     /etc/postfix/main.cf
myhostname = host.domain.tld
mydomain = domain.tld
# The mydomain parameter specifies the local internet domain name.
# The default is to use $myhostname minus the first component.
# This is used as a default value for many other configuration
# parameters.
mydomain = domain.tld

# SENDING MAIL
# The myorigin parameter specifies the domain that is appended
# to all local addresses that have no domain part.
myorigin = $myhostname
myorigin = $mydomain

# RECEIVING MAIL
# The inet_interfaces parameter specifies the network interface
# that the Postfix daemon uses to receive mail on by default.
# The software lists all active interfaces on the machine. The
# parameter also controls delivery of mail to user@ip.address.
# See also the proxy_interfaces parameter, for network addresses that
# act as relay or forwarder hosts.
# Note: you need to stop/start Postfix when this parameter changes.
inet_interfaces = all
inet_interfaces = $myhostname
inet_interfaces = $mydomain
inet_interfaces = localhost
inet_interfaces = all

# Enable IPv4, and IPv6 if supported
inet_protocols = ipv4
# The proxy_interfaces parameter specifies the network interface
# that the Postfix daemon receives mail on by default.
# proxy_interfaces = $inet_interfaces
proxy_interfaces = 1.1.1.4
proxy_interfaces = 1.1.1.4

```

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

```

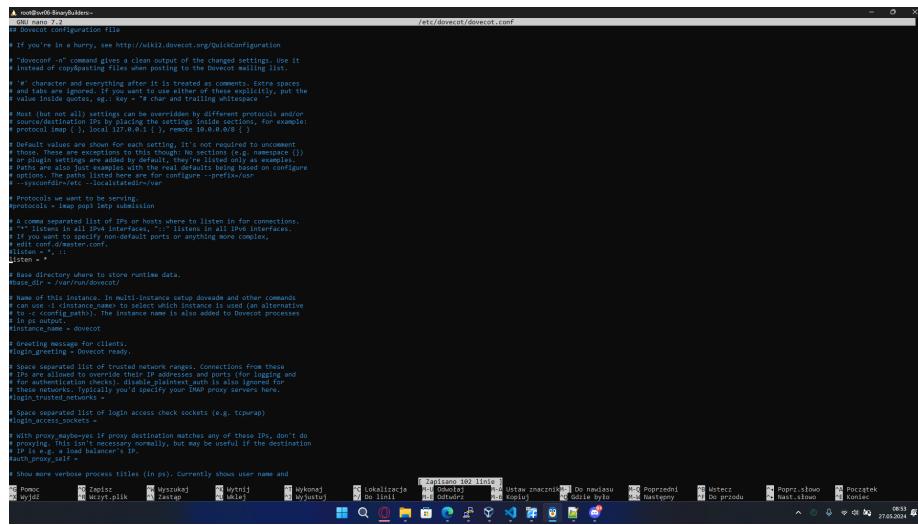
Test poczty jest dostępny tutaj.

## 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), or a list of words and/or
# source/destination IPs 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
# /var/lib/dovecot/ -var/dovecot/ -var/run/dovecot/
# /var/lib/dovecot/ -var/run/dovecot/
# Name of this instance. In multi-instance setup dovecot and other commands
# use < -c 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 local 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
# connections from these. Using this option you don't need proxy servers now.
# plugin_trusted_networks = 192.168.1.0/24

# Space separated list of login access check sockets (e.g. tcpwrap)
# auth_access = 192.168.1.10:12345

# If a client connects to proxy destination matches any of these IPs, don't do
# proxying. This isn't necessary normally, but may be useful if the destination
# is a proxy (e.g. a load balancer's IP).
# auth_proxy_self = 192.168.1.10

# Show more verbose process titles (in ps). Currently shows user name and
# process ID.
# auth_ps_title = %u %p

# Basic
# auth_mechanisms = plain login
# auth_mechanism_timeout = 60s
# auth_max_logins = 10
# auth_max光芒 = 1000
# auth_negative_ttl = 1 hour
# auth_cache_negative_ttl = 1 hour

# Space separated list of realms for SASL authentication mechanisms that need
# realm information. It's used when dovecot-sasl supports multiple realms.
# Many clients simply use the first one listed here, so keep no default realms
# auth_realms =

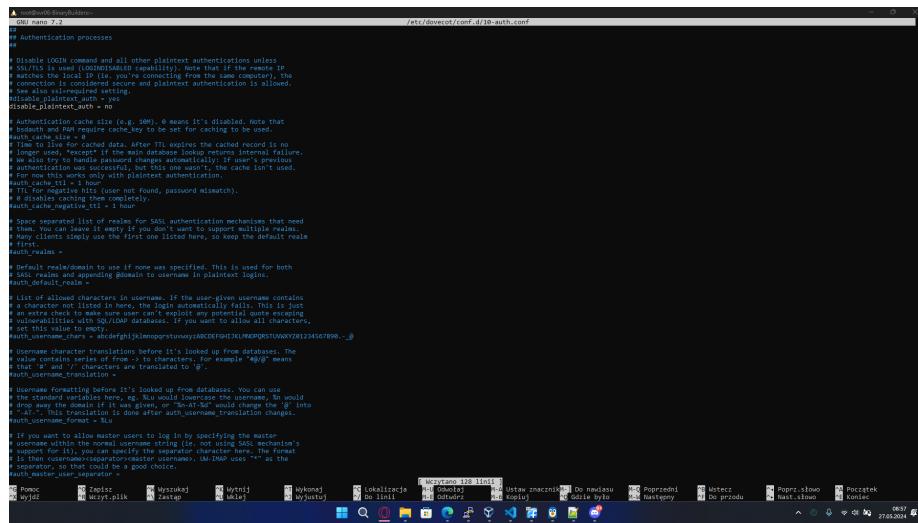
# Default realm/domain to use if none was specified. This is used for both
# SASL realms and appending #domain to usernames in plaintext logins.

# 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 doesn't mean it's allowed to have multiple characters
# in a row. If you want to allow multiple characters, use auth_username_translation
# instead. This is done after auth_username_translation changes.
# auth_username_charset = ^[\w\.\-\_]+

# 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), then you must enable this. The default value is "no". The
# reason is that colonname:password/master usernames. LM-IMAP uses ":" as the
# separator between colonname and password.
# auth_master_user_separator = :
```

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

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



```
# Authentication processes

# Disable LOGIN command and all other plaintext authentication unless
# SASL/PLAIN is used (LOGINDISABLED capability). Note that if the remote IP
# is not the same as the one you're connecting from (the client computer), the
# connection is considered secure and plaintext authentication is allowed.
# See also #allow_realm_list setting.
# disable_plaintext_auth = yes

# Authentication cache size (e.g. 10M). 0 means it's disabled. Note that
# SASL/PLAIN cache is shared with SASL/PLAIN-AES256-MD5. So if you want to
# use SASL/PLAIN-AES256-MD5, you should set its cache size to be set for caching
# SASL/PLAIN-AES256-MD5.
# auth_cache_size = 0

# Cache for negative hits (when user was not found). This is used
# for SASL/PLAIN-AES256-MD5. It's disabled by default. If you want to
# use it, you must enable SASL/PLAIN-AES256-MD5 first.
# auth_cache_negative_ttl = 1 hour
# auth_cache_negative_ttl = 1 hour

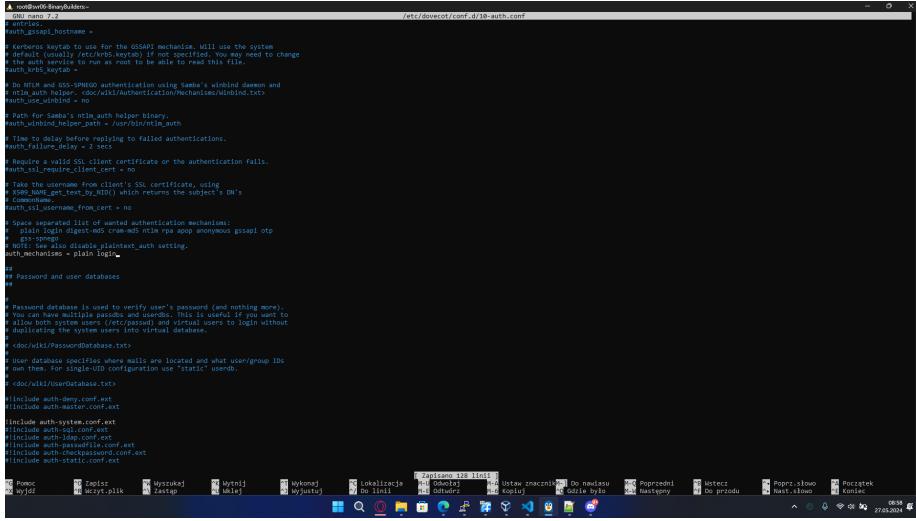
# Space separated list of realms for SASL authentication mechanisms that need
# realm information. It's used when dovecot-sasl supports multiple realms.
# Many clients simply use the first one listed here, so keep no default realms
# auth_realms =

# 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 doesn't mean it's allowed to have multiple characters
# in a row. If you want to allow multiple characters, use auth_username_translation
# instead. This is done after auth_username_translation changes.
# auth_username_charset = ^[\w\.\-\_]+

# 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), then you must enable this. The default value is "no". The
# reason is that colonname:password/master usernames. LM-IMAP uses ":" as the
# separator between colonname and password.
# auth_master_user_separator = :
```

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



```

# Dovecot 2.0+ configuration file - see /usr/share/doc/dovecot/README.Debian for details.
# root_gssapi_hostname = 127.0.0.1
# default (usually /etc/knobs/keytab) if not specified. You may need to change
# the auth service to run as root to be able to read this file.
# auth_knobs_path = /etc/knobs/auth

# Do NTLM and GSS-KERBEROS authentication using Seabba's winbind daemon and
# ntlm_auth helper. doc/wiki/AuthenticationMechanisms/Winbind.txt
# auth_ntlm_winbind = no
# auth_kerberos5 = no
# auth_kerberos5_helper_path = /usr/sbin/ntlm_otp
# auth_kerberos5_timeout = 30s
# Time to delay before replying to failed authentications.
# auth_failure_delay = 3 seconds
# Require a valid SSL client certificate or the authentication fails.
# auth_ssl_require_certificate = yes
# Take the username from client's SSL certificate, using
# X509_NAME_get_text_by_NSN which returns the subject's DN
# auth_ssl_username_from_cert = no
# Space separated list of wanted authentication mechanisms.
# plain login digest-md5 cram-md5 rpa apop anonymous gssapi otp
# Note: See also dovecot_plaintext_auth setting.
# auth_mechanisms = plain login
# auth_mechanisms = plain login

## Password and user databases
#  

#  

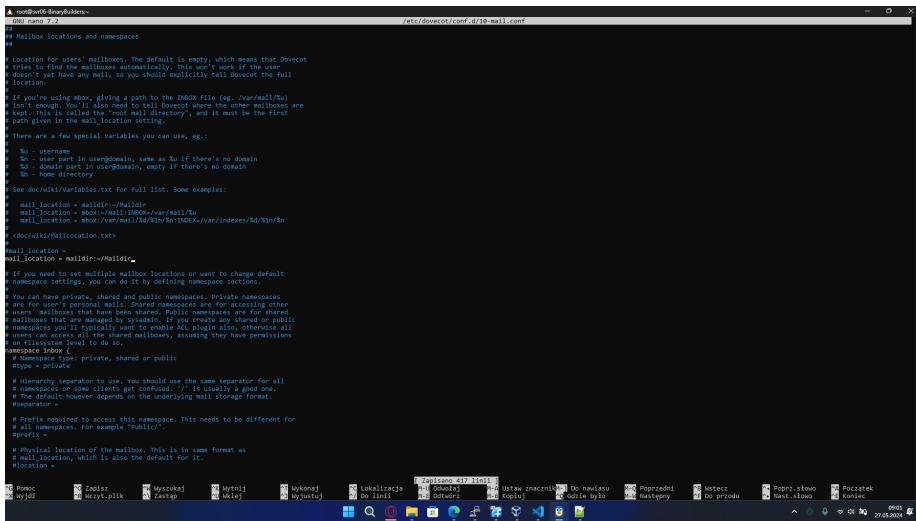
# Password database is used to verify user's password (and nothing more).
# It also can store user's additional information useful if you want to
# allow both system users (etc/passwd) and virtual users to login without
# duplicating the system users into virtual database.
# doc/wiki/PasswordDatabase.txt
# user_database specifies where mails are located and what user/group IDs
# one uses. For single-UID configuration use "static" userdb.
# doc/wiki/UserDatabase.txt
# include auth-system.conf.ext
# include auth-sql.conf.ext
# include auth-ldap.conf.ext
# include auth-passwdfile.conf.ext
# include auth-maildir.conf.ext
# include auth-static.conf.ext

```

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

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

**mail\_location = maildir:~/Maildir**



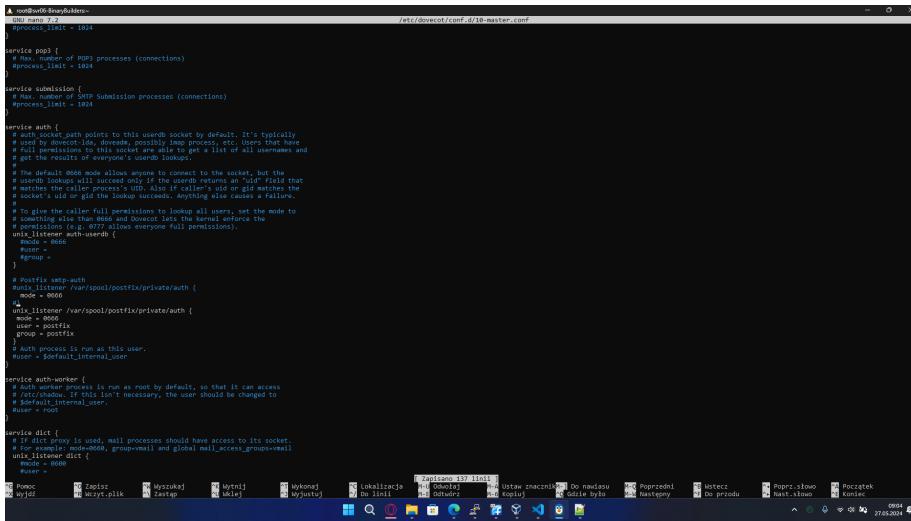
```

# Dovecot 2.0+ configuration file - see /usr/share/doc/dovecot/README.Debian for details.
# Mailbox locations and namespaces
# location for user's mailboxes. The default is empty, which means that Dovecot
# tries to find the mailboxes automatically. This won't work if the user
# doesn't yet have any mail, so you should explicitly tell Dovecot the full
# path to the mailbox.
# If you're using mbox, giving a path to the MBX file (e.g. /var/mail/%u)
# isn't enough. You'll also need to tell Dovecot where the other mailboxes are
# located. This is done by specifying a base directory, and it must be the first
# path given in the mail_location setting.
# There are a few special variables you can use, e.g.:
# %u - username
# %d - user part in user@domain, same as %u if there's no domain
# %h - home directory
# See doc/wiki/Variables.txt for full list. Some examples:
# mail_location = maildir:~/Maildir
# mail_location = maildir:/var/mail/%u
# mail_location = mbox:/var/mail/%u
# mail_location = maildir:/var/mail/indexes/%d/%u
# doc/wiki/Maillocation.txt
mail_location = maildir:~/Maildir
# If you need to set multiple mailbox locations or want to change default
# namespace settings, you can do so by defining namespace sections.
# Namespaces are for shared mailboxes. Public namespaces are for accessing
# other users' mailboxes that have been shared. Shared namespaces are for sharing
# your own mailboxes with other users. Private namespaces are for your
# own personal mailboxes. If you want to share your public namespaces with
# other users, you'll typically want to enable ACL plugin also, otherwise all
# users will have full access to them. You can also change any mail permissions
# on filesystem level to do so.
# # namespace type: private, shared or public
# type = private
# type = shared
# separator character to use. You should use the same separator for all
# namespaces or some clients get confused. '/' is usually a good one.
# # The default however depends on the underlying mail storage format.
# # prefix required to access this namespace. This needs to be different for
# # all namespaces. For example "Public".
# prefix =
# location = maildir:~/Maildir
# location 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:



```
# Dovecot - Dovecot - 7.2          /etc/dovecot/conf.d/10-master.conf
#process_limit = 1024

service pop3 {
    # Max. number of POP3 processes (connections)
    #process_limit = 1024
}

service submission {
    # Max. number of SMTP Submission processes (connections)
    #process_limit = 1024
}

service auth-worker {
    # Dovecot's auth-worker process is run as root by default, so that it can access
    # /etc/shadow. If this isn't necessary, the user should be changed to
    # something less privileged.
    #user = root
}

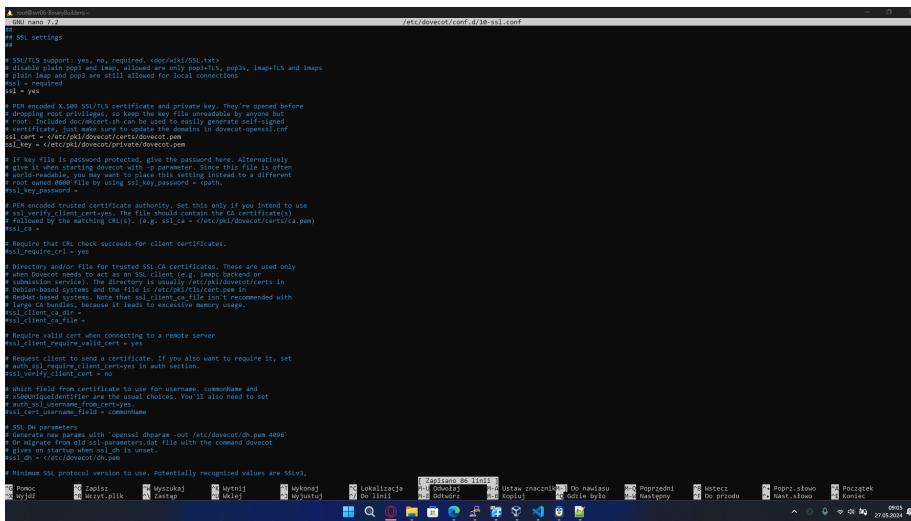
service auth-pwd-file {
    # SSL/TLS support is required.
    #ssl = yes
}

ssl = yes
```

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
```



```
# Dovecot - Dovecot - 7.2          /etc/dovecot/conf.d/10-ssl.conf
# SSL settings

ssl = yes

ssl/ltlsupport yes, no, required. <--> /etc/ssl/ltlsupport
# disable plain pop3 and imap, allowed are only pop3+tls, pop3s, imap+tls and imaps
# plain pop3 and imap are still allowed for local connections
#ssl = required

# If you need X.509 SSL/TLS certificate and private key, they're opened before
# dropping root privileges, so keep the key file unreadable by anyone but
# root. Included doc/meson.in can be used to easily generate self-signed
# certificates. See also https://gitlab.com/dovecot/dovecot/-/blob/main/doc/meson.in
#ssl_cert = /etc/pki/dovecot/certs/dovecot.pem
#ssl_key = /etc/pki/dovecot/certs/dovecot.pem

# If you have multiple certificates, you can list them here, alternatively
# give it when starting dovecot with -p parameter, since this file is often
# followed by a listing CRL(s). (e.g. ssl_ca </etc/pki/dovecot/certs/ca.pem>
#ssl_client_ca_file = /etc/pki/dovecot/certs/ca.pem

# Require that SSL check succeeds for client certificates.
#ssl_require_crl = yes

# Directory and/or file for trusted SSL CA certificates. These are used only
# when Dovecot needs to verify an SSL client (e.g. when
# ssl/ltlsupport is enabled). The directory is read from /etc/pki/dovecot/certs in
# Debian-based systems and the file is /etc/ssl/certs/client.pem in
# others. It's recommended to use a single large CA bundle, because it leads to excessive memory usage.
#ssl_client_ca_file = /etc/ssl/certs/ca-certificates.crt

# Require valid cert when connecting to a remote server
#ssl_client_require_valid_cert = yes

# Require client to present a certificate, if you also want to require it, set
# auth_pvtkey_file and client_ca_file in auth section.
#ssl_verify_client_cert = no

# Which field from certificate to use for username, commentname and
# commentdomain. These are used for SASL password field choices. You'll need to set
# auth_username_field and commentname_field + commentdomain
#ssl_username_field = commentname

# SSL/TLS parameters
#ssl_migrate_from /etc/dovecot/ssl-parameters.dat file with the command dovecot
# ssl-migrate when dovecot is runed
#ssl_dh = /etc/dovecot/dh.pem

# Minimum SSL protocol version to use. Potentially recognized values are SSLv3,
```

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
```

```
[root@v80-binarybuilders ~]# systemctl enable --now dovecot
[root@v80-binarybuilders ~]# firewall-cmd --add-service={pop3,imap} --permanent
[root@v80-binarybuilders ~]# firewall-cmd --reload
[root@v80-binarybuilders ~]#
```

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:

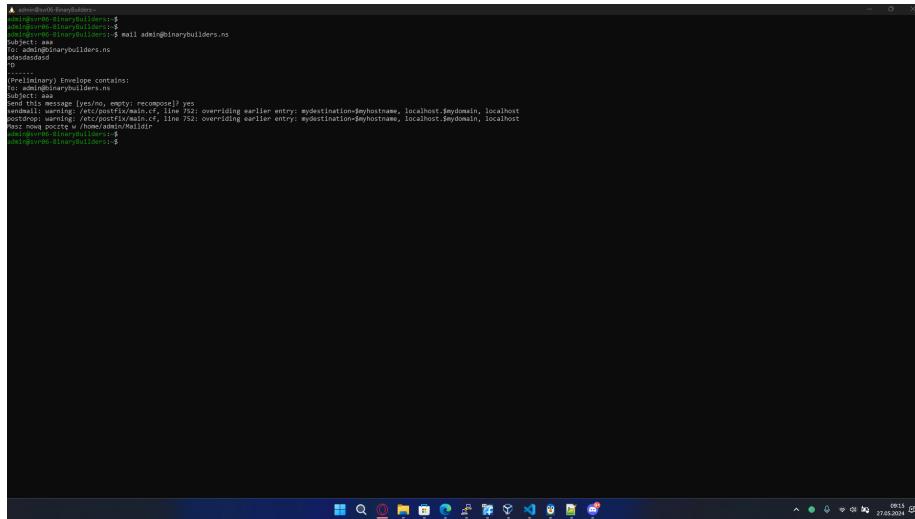
```
export MAIL=$HOME/Maildir
```

Następnie weryfikuje działanie powyższego skryptu:

```
[root@v80-binarybuilders ~]# sudo dnf install mailx -y
[root@v80-binarybuilders ~]# nano /etc/profile.d/mail.sh
[root@v80-binarybuilders ~]# source /etc/profile
[root@v80-binarybuilders ~]# echo $MAIL
[root@v80-binarybuilders ~]# mail
[root@v80-binarybuilders ~]# exit
[root@v80-binarybuilders ~]#
```

Rysunek 93: Test zmiennej środowiskowej MAIL

Przykład użycia tekstowego klienta poczty na serwerze:



```
Administrator:~$ mailx
Administrator:~$ mail admin@binarybuilders.ns
Administrator:~$ mail admin@binarybuilders.ns
To: admin@binarybuilders.ns
cc: 
Administrator:~$ Preliminary Envelope contains:
To: admin@binarybuilders.ns
Administrator:~$ Send this message [yes/no, empty: recompose]? yes
Administrator:~$ Postdrop warning: /etc/postfix/main.cf, line 752: overriding earlier entry: mydestination=$myhostname, localhost $mydomain, localhost
Administrator:~$ Postdrop warning: /etc/postfix/main.cf, line 752: overriding earlier entry: mydestination=$myhostname, localhost $mydomain, localhost
Administrator:~$ New message for user admin in /home/admin/Maildir
Administrator:~$
```

Rysunek 94: mailx – przykład użycia

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

#### 4.18 Dodatkowa usługa git (jako serwer) – instalacja i konfiguracja

Aby zainstalować git na serwerze należy wydać polecenie:

```
sudo dnf install git -y
```

W kolejnym kroku trzeba wykonać następujące komendy:

```
su - git
mkdir ~/.ssh
chmod 700 ~/.ssh
touch ~/.ssh/authorized_keys
chmod 600 ~/.ssh/authorized_keys
mkdir repozytoria
cd repozytoria/
mkdir BinaryBuilders.git
cd BinaryBuilders.git
repozytoria/BinaryBuilders.git$ git init --bare
systemctl restart ssh
```

Wytłumaczenie tych komend:

- su - git

Komenda ta oznacza "przełącz użytkownika" (su) na użytkownika o nazwie git. Flaga - oznacza, że chcemy przełączyć się na użytkownika git z ustawieniami jego środowiska.

- mkdir /.ssh

Tworzy nowy katalog o nazwie .ssh w katalogu domowym użytkownika git. Ten katalog będzie używany do przechowywania kluczy SSH. Następnie chmod 700 /.ssh ustawia uprawnienia (prawa dostępu) dla tego katalogu na 700, co oznacza, że tylko właściciel (użytkownik git) ma pełny dostęp do tego katalogu, a inni nie mają dostępu do jego zawartości.

- touch /.ssh/authorized\_keys

Tworzy pusty plik o nazwie authorized\_keys w katalogu .ssh. Ten plik będzie przechowywał klucze publiczne SSH, które pozwolą na uwierzytelnianie użytkowników. Następnie chmod 600 /.ssh/authorized\_keys ustawia uprawnienia dla pliku authorized\_keys na 600, co oznacza, że tylko właściciel ma prawo do odczytu i zapisu do tego pliku, a inni nie mają żadnych uprawnień dostępu.

- mkdir repozytoria

Tworzy nowy katalog o nazwie repozytoria. To może być katalog główny, w którym będą przechowywane wszystkie repozytoria Git.

- cd repozytoria/

Przechodzi do katalogu repozytoria.

- mkdir BinaryBuilders.git

Tworzy nowy katalog o nazwie BinaryBuilders.git. Ta nazwa sugeruje, że to będzie repozytorium Git, ale nazwy z .git na końcu są konwencją dla repozytoriów Git, które są "gołe" lub "bezwarstwowe" (bare), co oznacza, że nie zawierają pracy na kodzie, tylko same dane repozytorium.

- cd BinaryBuilders.git

Przechodzi do katalogu BinaryBuilders.git.

- git init --bare

Inicjuje nowe repozytorium Git w trybie "bezwarstwowym" (bare), co oznacza, że będzie to repozytorium, które nie będzie zawierać żadnej pracy na kodzie, a jedynie historię repozytorium i dane kontrolne Git.

- systemctl restart ssh

Uruchamia ponownie usługę SSH na serwerze. To polecenie może być potrzebne, jeśli dokonano jakichkolwiek zmian w konfiguracji SSH lub kluczach, aby zastosować te zmiany.

W kolejnym kroku wykonuje następujące polecenia na windows (łącząc się z serwerem po karcie enp0s8, gdyż ustawiona ona jest na sieć mostkowaną):

```
ssh-keygen
scp "C:\Users\Maciej\.ssh\id_rsa.pub"
→ git@192.168.1.100:~/.ssh/authorized_keys
ssh git@192.168.1.100
Poniżej komendy do testowania:
git clone git@192.168.1.100:~/repozytoria/BinaryBuilders.git
echo abc > test.txt
git add .; git commit -m "Pierwszy commit"; git push
```

Wytyłumaczenie tych komend:

- scp "C:/Users/Maciej/.ssh/id\_rsa.pub" git@192.168.1.100:~/.ssh/authorized\_keys  
Polecenie scp kopiuje klucz publiczny (id\_rsa.pub) z lokalnego komputera do katalogu ~/.ssh/authorized\_keys na serwerze Git. Ten krok umożliwia uwierzytelnianie się na serwerze Git za pomocą klucza SSH.
- ssh git@192.168.1.100  
To polecenie próbuje nawiązać połączenie SSH z serwerem Git, aby sprawdzić, czy klucz SSH został poprawnie dodany i czy możesz się zalogować na serwerze Git za pomocą klucza SSH.
- git clone git@192.168.1.100:~/repozytoria/BinaryBuilders.git  
Klonuje repozytorium Git znajdujące się na serwerze Git pod wskazaną ścieżką ~/repozytoria/BinaryBuilders.git.
- cd ./BinaryBuilders  
Przechodzi do katalogu BinaryBuilders, gdzie znajduje się lokalne repozytorium Git.
- echo abc > test.txt  
Tworzy nowy plik tekstowy o nazwie test.txt i umieszcza w nim jedną linię tekstu zawierającą słowo "abc".

Efekt działań opisanych powyżej Wynik git można znaleźć [tutaj](#).

```
Windows PowerShell
PS C:\Users\Maciej\Desktop\GIT\BinaryBuilders> ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (C:/Users/Maciej/.ssh/id_rsa):
C:/Users/Maciej/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
The authenticity of this host has been saved in C:/Users/Maciej/.ssh/id_rsa
Your public key has been saved in C:/Users/Maciej/.ssh/id_rsa.pub
You have E-mail address:
maciej@Maciej-gabryatte
The key's randomart image is:
+---[RSA 2048]---+
|          . |
|          . |
|          . |
|          . |
|          . |
|          . |
|          . |
|          . |
|          . |
|          . |
+---[SHA256]---+
PS C:\Users\Maciej\Desktop\GIT\BinaryBuilders> scp "C:/Users/Maciej/.ssh/id_rsa.pub" git@192.168.1.100:/.ssh/authorized_keys
The authenticity of host '192.168.1.100' ('192.168.1.100') can't be established.
ED5519 key fingerprint is SHA256:Wk99nQVJCI40McNghpsH1RMK0KO9cess10mBwU4X0.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
Warning: Permanently added '192.168.1.100' (ED5519) to the list of known hosts.
git@192.168.1.100's password:
id_rsa.pub                                              100% 577 185.4KB/s  00:00
PS C:\Users\Maciej\Desktop\GIT\BinaryBuilders> ssh git@192.168.1.100
Last login: Sun Jun 2 23:53:22 2024
[git@192.168.1.100 ~]$ logout
[git@192.168.1.100 ~]$
```

Rysunek 95: git – konfiguracja

## 5 Testy działania wdrożonych usług

## 5.1 DNS

```
[Maciej-MacBook-Pro: ~] $ ./poceta BinaryBuilders.ns
Plik Edyjny Wskaźnik Wyświetlacz Terminal Pomoc
maciej@maciej-VirtualBox: ~
```

Maciej@maciej-VirtualBox: ~\$ dig poceta.BinaryBuilders.ns

```
<> ;迪 G 9.18.18-Debian-22.04.2-Ubuntu <>> poceta.BinaryBuilders.ns
;; Got answer:
;; ->HEADER<< opcode: QUERY, status: NOERROR, id: 60723
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
;; OPT PSEUDORECORDS:
;; EDNS: version: 0, flags: udp: 65494
; poceta.BinaryBuilders.ns. IN A
;; ANSWER SECTION:
poceta.BinaryBuilders.ns. 7120 IN A 192.168.230.1

Maciej@maciej-VirtualBox: ~$ dig www.BinaryBuilders.ns
<> ;迪 G 9.18.18-Debian-22.04.2-Ubuntu <>> www.BinayBuilders.ns
;; global options: +cmd
;; Got answer:
;; ->HEADER<< opcode: QUERY, status: NOERROR, id: 47558
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
;; OPT PSEUDORECORDS:
;; EDNS: version: 0, flags: udp: 05494
; www.BinayBuilders.ns. IN A
;; ANSWER SECTION:
www.BinayBuilders.ns. 7101 IN A 192.168.230.1

;; Query time: 4 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Fr May 24 22:51:59 CEST 2024
;; MSG SIZE rcvd: 69
```

Maciej@maciej-VirtualBox: ~\$ ping -c 2 poceta

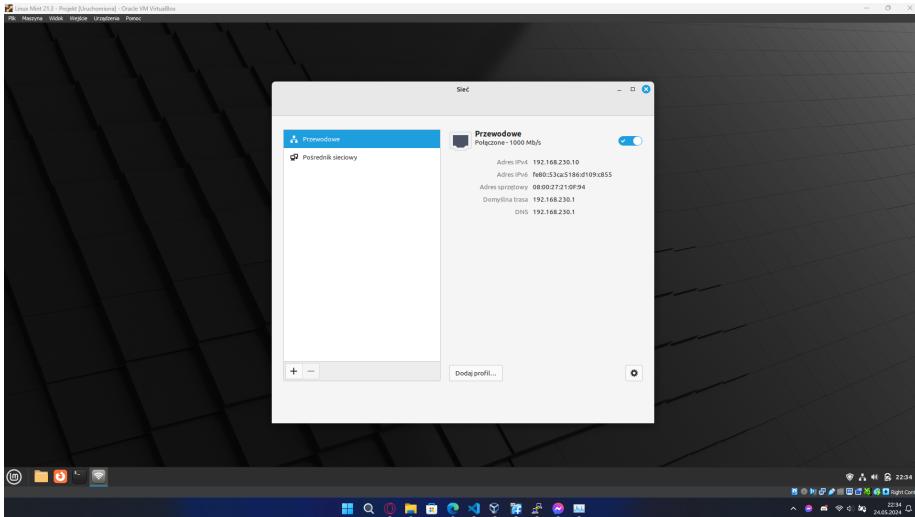
```
PING poceta.BinaryBuilders.ns [192.168.230.1] 56(84) bytes of data.
64 bytes from svr02.poceta.BinaryBuilders.ns [192.168.230.1]: icmp_seq=1 ttl=64 time=1.46 ms
64 bytes from svr02.poceta.BinaryBuilders.ns [192.168.230.1]: icmp_seq=2 ttl=64 time=26.38 ms

--- poceta.BinaryBuilders.ns ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1000ms
rtt min/avg/max/mdev = 1.466/13.358/1.466/0.102 ms
```

Rysunek 96: 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.

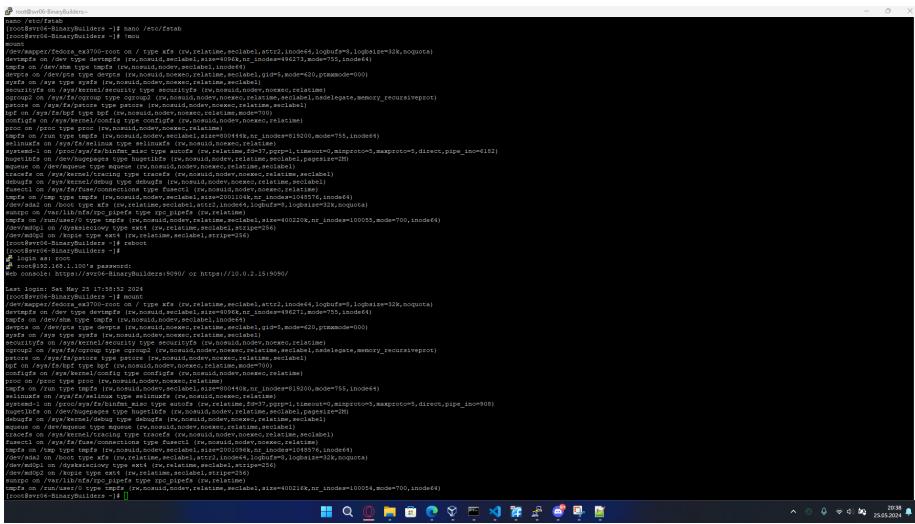
## 5.2 DHCP



Rysunek 97: Instalacja DHCP

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

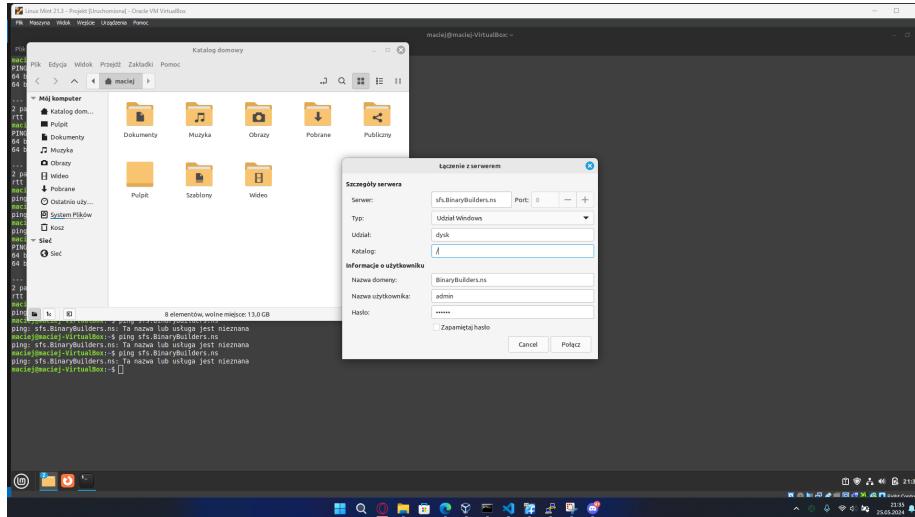
## 5.3 Raid 5



Rysunek 98: Test automatycznego montowania partycji po ponownym uruchomieniu serwera

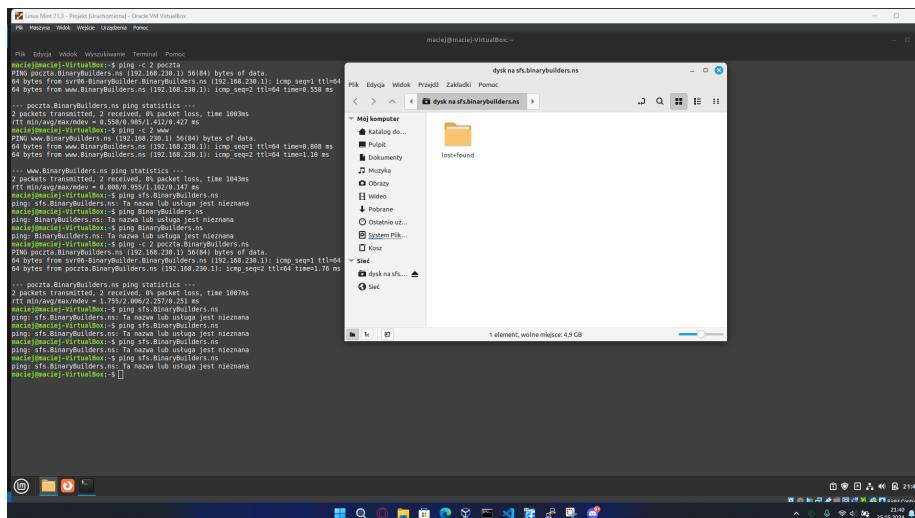
Jak widać na powyższym zrzucie ekranu po restarcie serwera partycje są montowane poprawnie.

## 5.4 Samba



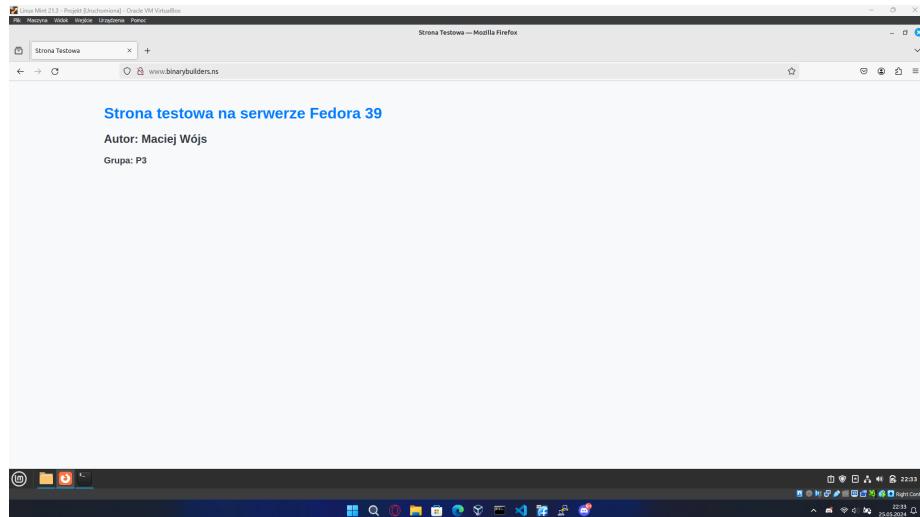
Rysunek 99: Samba – próba podłączenia się do udziału na serwerze

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 Udziały Windows, ostatecznie należy wypełnić wymagane dane. Przykładowa próba podłączenia powyżej, a efekt tego działania poniżej.



Rysunek 100: Samba – wynik poprzedniego kroku

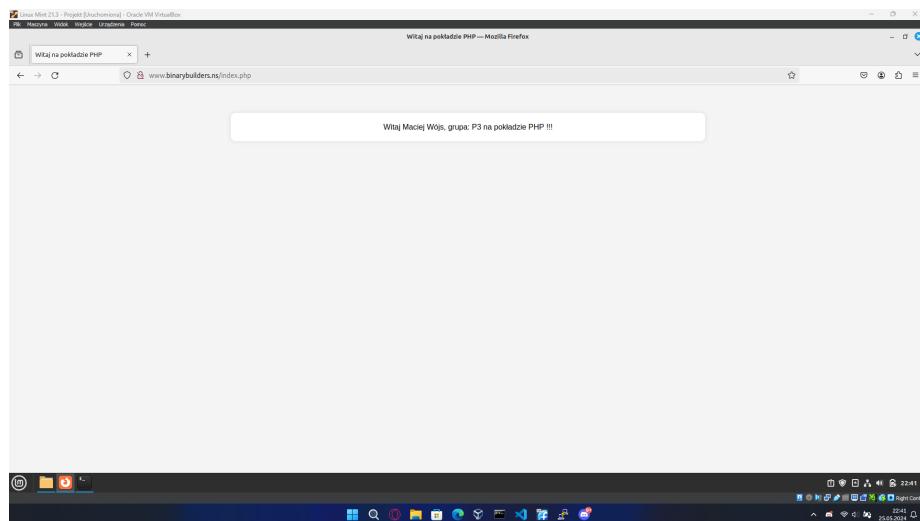
## 5.5 HTTP



Rysunek 101: Test działania serwera WWW

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

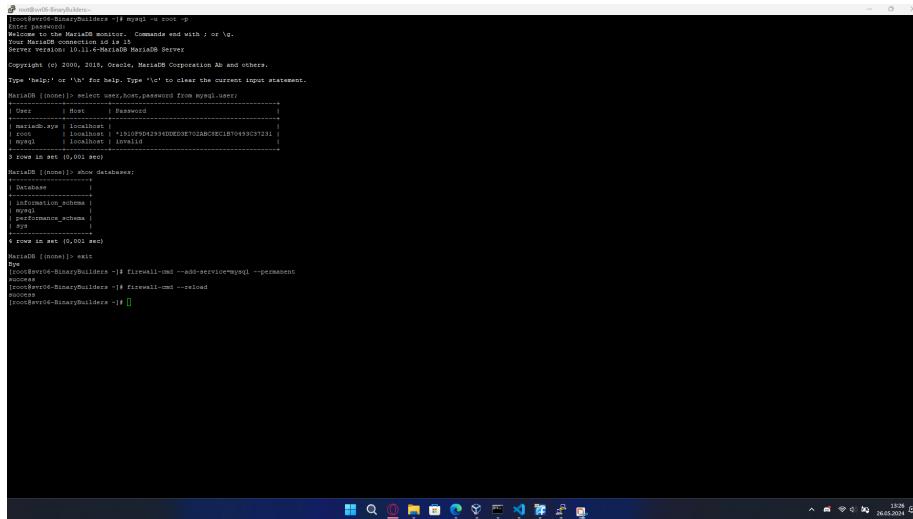
## 5.6 PHP



Rysunek 102: 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@srv04-BinaryBuilders ~]# mysql -u root -p
Enter password:
Your MySQL connection id is 15
Server version: 10.1.14-MariaDB MariaDB Server

Copyright (c) 2000, 2016, 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          |
+-----+-----+-----+
| mariadb_root | localhost |              |
| mariadb_root | %        |              |
| mariadb_root | %        |              |
| mariadb_root | localhost |              |
| mysql      | localhost |              |
| mysql      | %        |              |
| mysql      | %        |              |
| sys       | localhost |              |
+-----+-----+-----+
3 rows in set (0,001 sec)

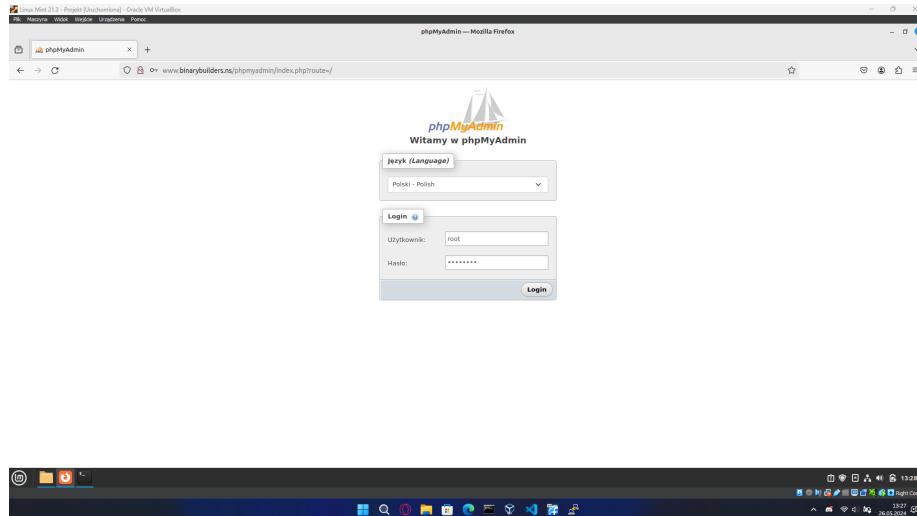
MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys               |
+--------------------+
4 rows in set (0,001 sec)

MariaDB [(none)]> exit
Bye
[root@srv04-BinaryBuilders ~]# firewall-cmd --add-service=mysql --permanent
success
[root@srv04-BinaryBuilders ~]# firewall-cmd --reload
success
[root@srv04-BinaryBuilders ~]#
```

Rysunek 103: Test usługi mariadb (MySQL)

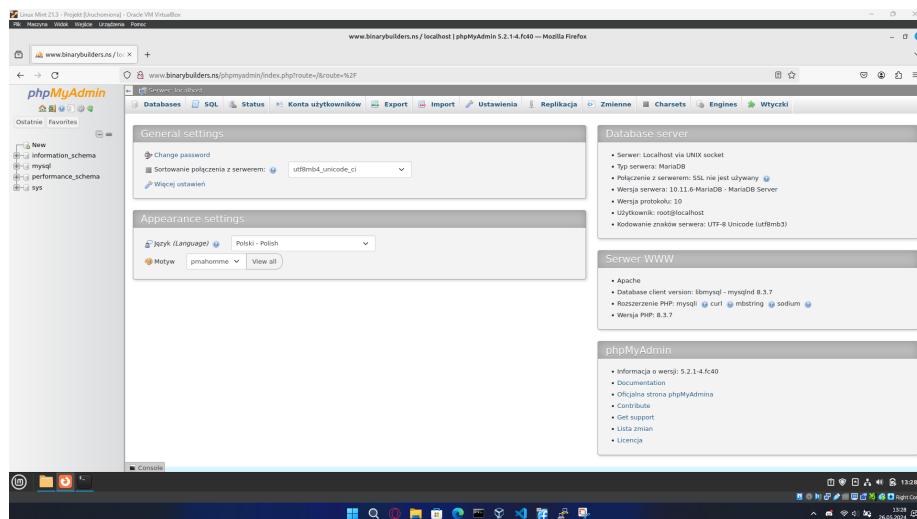
Jak widać podłączenie zz bazą danych działa. Następnym i ostatnim krokiem jest dodanie usługi MySQL do dozwolonych usług w zaporze ogniewej.

## 5.8 phpMyAdmin



Rysunek 104: Test usługi phpMyAdmin – część pierwsza

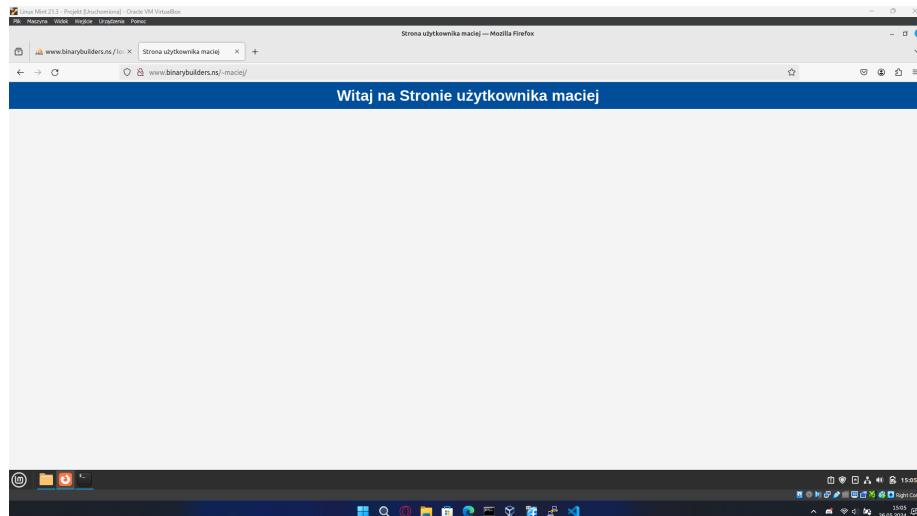
Pierwszym krokiem w testowaniu phpMyAdmin jest otworzenie strony serwera tej usługi w przeglądarce a następnie zalogowanie się na konto.



Rysunek 105: Test usługi phpMyAdmin – część druga

Jak widać po zalogowaniu mamy dostęp do administrowania bazami danych.

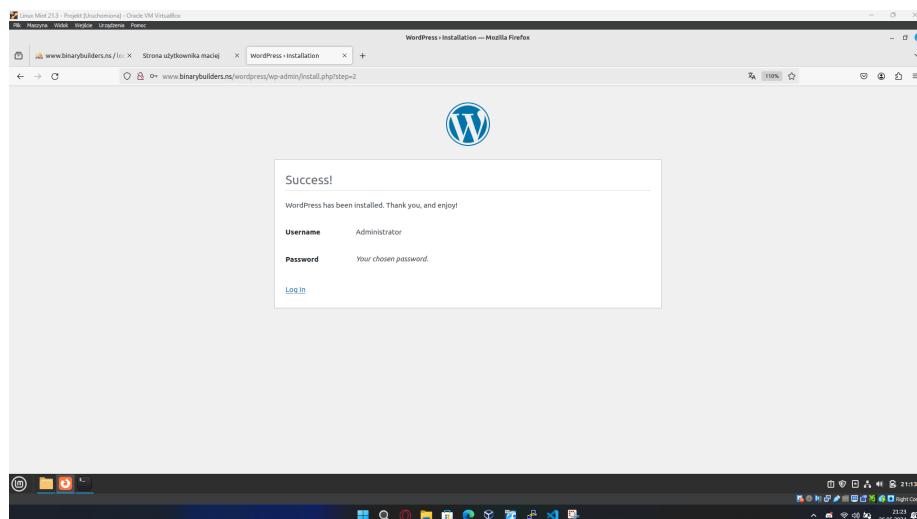
## 5.9 UserDir – serwer http



Rysunek 106: MySQL – instalacja część druga

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

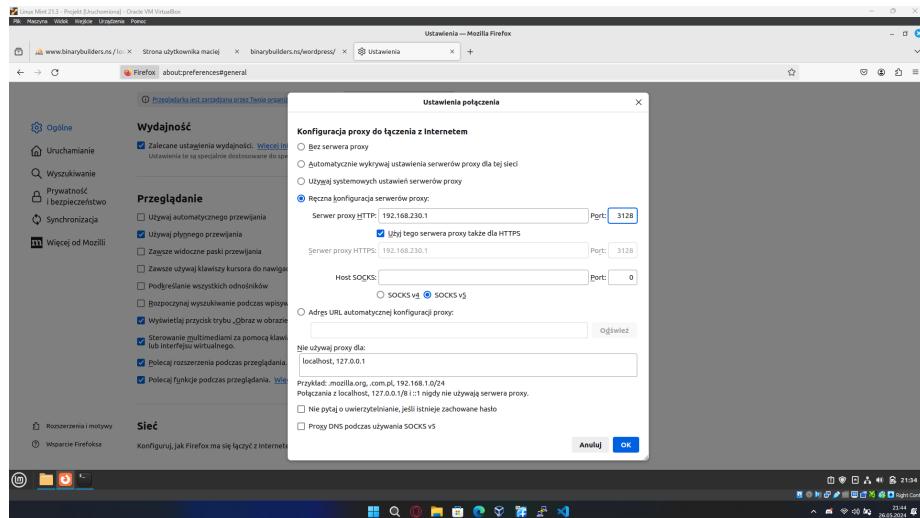
## 5.10 WordPress



Rysunek 107: Dashboard WordPress'a

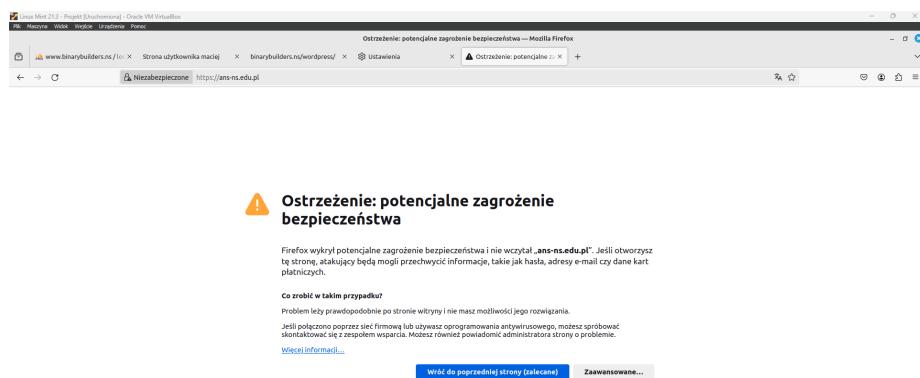
Jak widać powyżej po zalogowaniu się na konto widać dashboard WordPress'a, skąd można skonfigurować stronę internetową.

## 5.11 Proxy



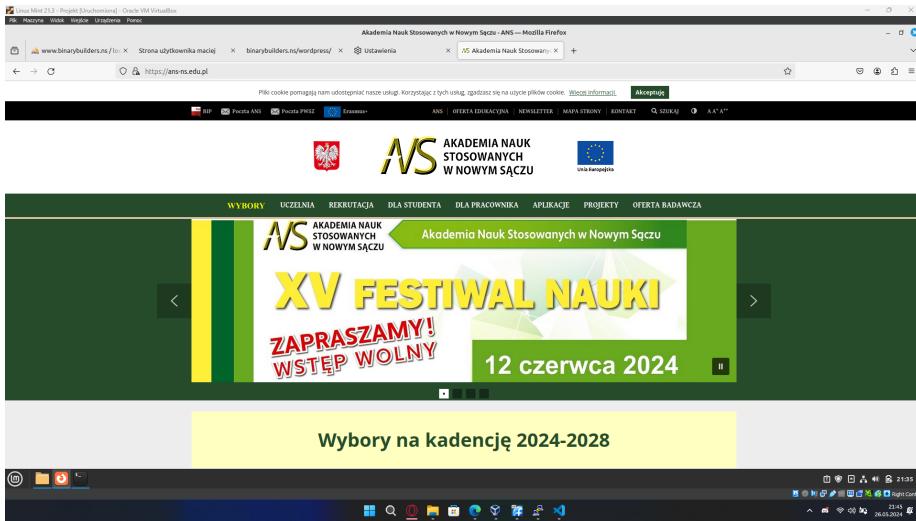
Rysunek 108: Proxy – ustawienie w FireFox

Konfiguracja proxy w przeglądarce FireFox.



Rysunek 109: Proxy – dostęp do strony ostrzelenie

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



Rysunek 110: Proxy – wynik strony po zignorowaniu ostrzerazenia

Widok strony po zignorowaniu ostrzerazenia.

```
#!/usr/bin/python3
# This script is designed to ignore certain IP addresses from being
# blocked by squid. It reads a file containing IP ranges and
# adds them to the /etc/squid/squid.conf file under the 'except' section.

# Read the file containing IP ranges
with open('ignore_ip_ranges.txt') as f:
    ignore_ips = f.read()

# Write the ignore IPs to the squid.conf file
with open('/etc/squid/squid.conf', 'a') as f:
    f.write(ignore_ips)

# Reload squid to apply changes
os.system('sudo systemctl reload squid')

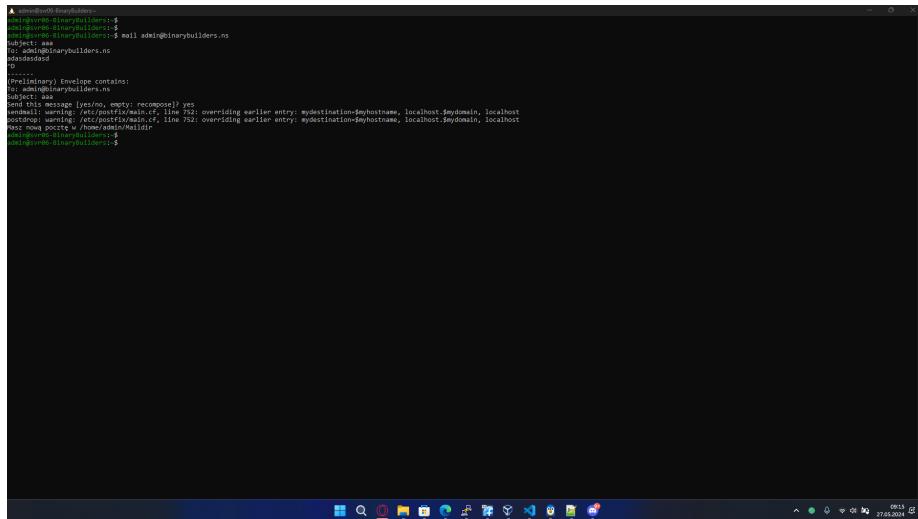
# Monitor network traffic using netstat -an
# You can use 'tail -f' to see real-time updates
os.system('netstat -an | grep "ESTABLISHED"')
```

Rysunek 111: 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. Linijki 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"
```

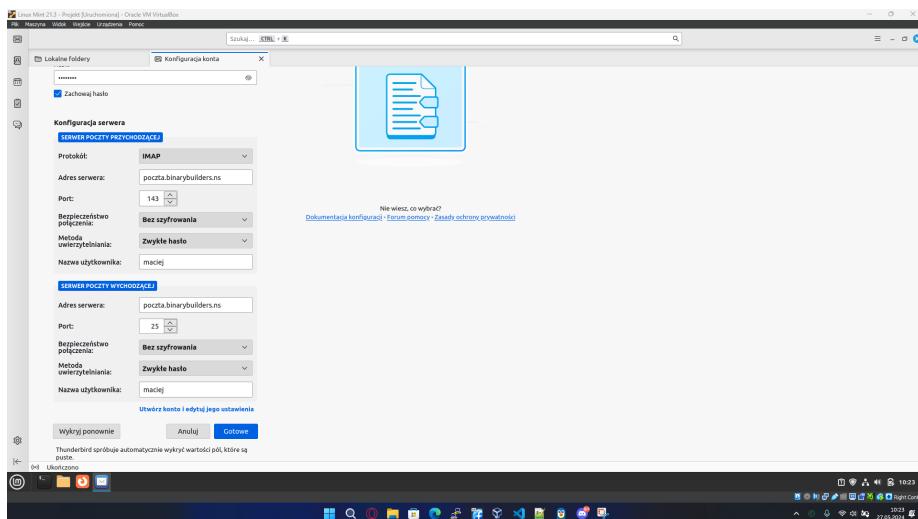
## 5.12 Poczta



```
Administrator:~ binarybuilders:~$ ls -l /etc/postfix/main.cf
Administrator:~ binarybuilders:~$ mail admin@binarybuilders.ns
Administrator:~ binarybuilders:~$ To: admin@binarybuilders.ns
Administrator:~ binarybuilders:~$ Subject: test
Administrator:~ binarybuilders:~$ send this message [yes/no, empty: reccognize] yes
Administrator:~ binarybuilders:~$ /etc/postfix/main.cf: line 752: overriding earlier entry: mydestination=$myhostname, localhost.$mydomain, localhost
Administrator:~ binarybuilders:~$ postdrop: warning: /etc/postfix/main.cf, line 752: overriding earlier entry: mydestination=$myhostname, localhost.$mydomain, localhost
Administrator:~ binarybuilders:~$ test now почты в /home/admin/.Maildir
Administrator:~ binarybuilders:~$ ls -l /home/admin/.Maildir
Administrator:~ binarybuilders:~$
```

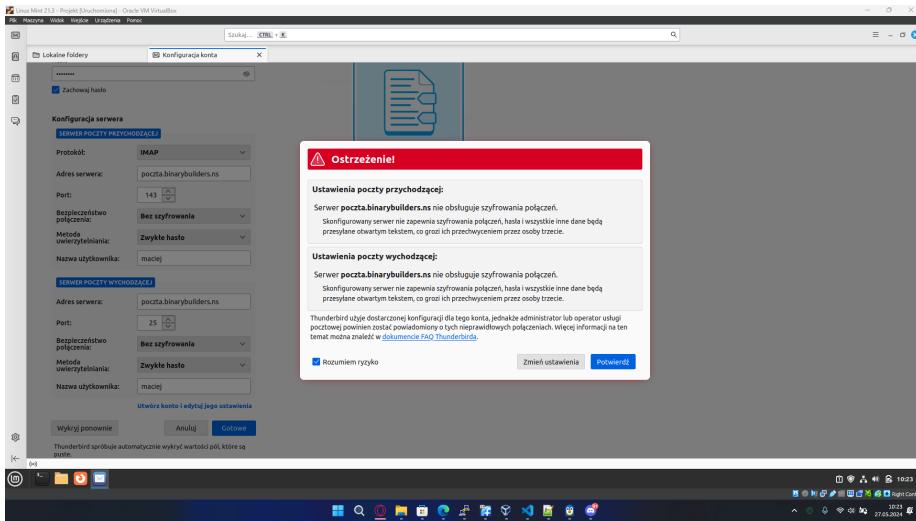
Rysunek 112: Wysłanie maila na serwerze

Na zrzucie powyżej przedstawiony jest przykład jak korzystać z klienta tekstuowego poczty na serwerze.



Rysunek 113: Thunderbird - konfiguracja

Konfiguracja programu Thunderbird – klienta poczty.



Rysunek 114: Wysłanie maila na serwerze

Na zdjęciu powyżej przedstawione jest ostrzeżenie dotyczące bezpieczeństwa – konfiguracja serwera nie obsługuje szyfrowania.

### 5.13 Git

```

PS C:\Users\Maciej\Desktop\GIT> git clone git@192.168.1.100:~/repozytoria/BinaryBuilders.git
Cloning into 'BinaryBuilders'...
warning: You appear to have cloned an empty repository.
PS C:\Users\Maciej\Desktop\GIT> cd BinaryBuilders
PS C:\Users\Maciej\Desktop\GIT\BinaryBuilders> echo abc > test.txt
PS C:\Users\Maciej\Desktop\GIT\BinaryBuilders> git add .
PS C:\Users\Maciej\Desktop\GIT\BinaryBuilders> git commit -m "Pierwszy commit"
[master (root-commit) cbc959c] Pierwszy commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 test.txt
PS C:\Users\Maciej\Desktop\GIT\BinaryBuilders>
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 230 bytes | 230.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To 192.168.1.100:~/repozytoria/BinaryBuilders.git
 * [new branch]      master -> master
PS C:\Users\Maciej\Desktop\GIT\BinaryBuilders>
  
```

Rysunek 115: git – konfiguracja

Na zrzucie ekranu przedstawione jest klonowanie repozytorium, tworzenie commita oraz wypchnięcie kodu na serwer.

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

FFFFFFFFFFFFFFn

## **7 Wnioski**

EEEEEEEEE

## 8 Literatura

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