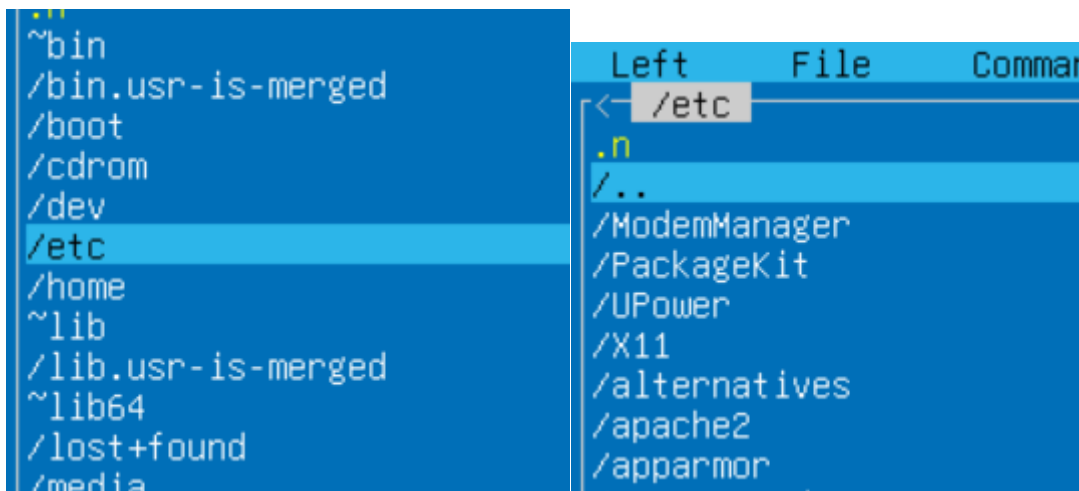


Konfiguracja Apache 2

1. Instalujemy apache2 poleceniem: **# apt install apache2**
2. Strona domyślna powitalna serwera apache2 znajduje się w katalogu: `/var/www/html`. Jest ona powiązana z plikiem **000-default.conf** z katalogu `/sites-available`. W celu zmiany strony powitalnej na inną kasujemy `index.html` i tworzymy własny.
3. Apache2 instaluje się w katalogu `/etc`:



4. Struktura Apache2 jest następująca:



5. W katalogu `/sites-available` przygotowuję wzorcowy plik na bazie **000-default.conf**.

6. Kasuję wszystkie #. Zostawiam elementy potrzebne mi do konfiguracji hostów wirtualnych.

```
ls -l /etc/apache2/sites-available
-rw-r--r-- 1 root root 1024 2017-07-14 10:10 000-default.conf
-rw-r--r-- 1 root root 1024 2017-07-14 10:10 default-ssl.conf
```

- Przykładowy host wirtualny wygląda następująco:

```
/etc/apache2/sites-available/000-default.conf [----] 3
<VirtualHost *:80>
<----->
<----->ServerName example.com
        ServerAlias www.example.com_

<----->ServerAdmin webmaster@localhost
<----->DocumentRoot /var/www/html

<----->#LogLevel info ssl:warn

<----->ErrorLog ${APACHE_LOG_DIR}/error.log
<----->CustomLog ${APACHE_LOG_DIR}/access.log combined

<----->#Include conf-available/serve-cgi-bin.conf
</VirtualHost>
```

7. Na podstawie pliku 000-default.conf tworzę plik hosta wirtualnego.

```
root@server:/etc/apache2/sites-available# ls
000-default.conf  default-ssl.conf
root@server:/etc/apache2/sites-available# cp 000-default.conf domena.org.conf
```

8. Plik hosta wirtualnego musi kończyć się na **.conf**. Początek to jednak nazwa domeny. Przykłady plików: dell.com.conf, sejm.gov.conf. Plik domeny po konfiguracji wygląda następująco:

```

/etc/apache2/sites-available/domena.org.conf  [-M--]  0
<VirtualHost *:80>
<----->ServerName domena.org
        ServerAlias www.domena.org

<----->ServerAdmin webmaster@domena.org
<----->DocumentRoot /var/www/domena.org

<----->#LogLevel info ssl:warn
<----->ErrorLog ${APACHE_LOG_DIR}/error.log
<----->CustomLog ${APACHE_LOG_DIR}/access.log combined
<----->#Include conf-available/serve-cgi-bin.conf
</VirtualHost>

```

9. W odpowiednim katalogu u nas /var/www tworzymy odpowiedni katalog ze stroną (index.html). Katalog domeny w /var/www powinien odpowiadać temu nazwie domenowej.

- Gdy posiadamy następujące zależności:

```

/etc/apache2/sites-available/domena.org.conf  [-M--]  0
<VirtualHost *:80>
<----->ServerName domena.org
        ServerAlias www.domena.org

<----->ServerAdmin webmaster@domena.org
<----->DocumentRoot /var/www/domena.org

<----->#LogLevel info ssl:warn
<----->ErrorLog ${APACHE_LOG_DIR}/error.log
<----->CustomLog ${APACHE_LOG_DIR}/access.log combined
<----->#Include conf-available/serve-cgi-bin.conf
</VirtualHost>

```

```

root@server:/etc/apache2/sites-available# cd /var/www
root@server:/var/www# mkdir domena.org

```

```

root@server:/var/www# ls
domena.org  html
root@server:/var/www# cd domena.org
root@server:/var/www/domena.org# pico index.html

```

10. Uruchamiamy stronę wydając komendę: **# a2ensite domena.org.conf**

```

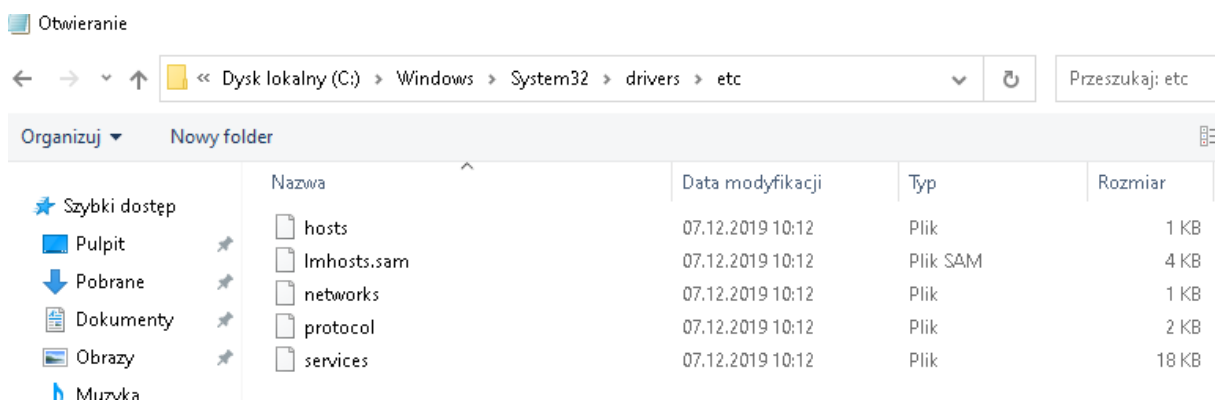
root@server:/var/www/domena.org# a2ensite domena.org.conf
Enabling site domena.org.
To activate the new configuration, you need to run:
  systemctl reload apache2
root@server:/var/www/domena.org#

```

11. Wydajemy komendę apache2: **# systemctl reload apache2** lub wydając: **#/etc/init.d/apache2 reload**.

12. Testowanie strony:

- połączenie naszej maszyny wirtualnej z Windows. W Windows edycja pliku: hosts. Edytuję jako Administrator plik ukryty w:



```

#
#       102.54.94.97       rhino.acme.com           # source server
#       38.25.63.10      x.acme.com               # x client host
10.2.100.100    domena.org

```

- instalacja w linux programu przeglądarki internetowej: lynx oraz konfiguracja pliku **/etc/hosts** tak by wskazywał na naszą domenę.

```

root@server:/var/www/domena.org# apt install lynx
Reading package lists... Done
Building dependency tree... Done

```

```

/etc/hosts      [-M--] 12
127.0.0.1 localhost
127.0.1.1 server
10.2.100.100 domena.org

```

```

URL to open: domena.org
Arrow keys: Up and Down to move. Right to follow a link; Left to go back.
H)elp O)ptions P)rint G)o M)ain screen Q)uit /=search [delete]=history list

```

```
Data transfer complete
Arrow keys: Up and Down to
H)elp O)ptions P)rint G)o
```

Uruchomienie strony w katalogu domowym użytkownika

1. Tworzymy użytkowników: adduser uzytkownik
2. Każdy użytkownik w swoim domowym katalogu, najczęściej /home/* tworzy katalog: **public_html** zgodnie z modułem: **userdir** apache2

```
<- /etc/apache2
.n
/..
/conf-available
/conf-enabled
/mods-available
/mods-enabled
/sites-available
/sites-enabled
apache2.conf
envvars
magic
ports.conf
```

Moduł wygląda następująco:

```
/etc/apache2/mods-available/userdir.conf [----] 0 L:[ 1+ 0 1/ 9] >
UserDir public_html
UserDir disabled root

<Directory /home/*/public_html>
<----->AllowOverride FileInfo AuthConfig Limit Indexes
<----->Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
<----->Require method GET POST OPTIONS
</Directory>
```

3. W katalogu **public_html** użytkownik tworzy stronę (index.html)
- dobrze by nadał prawa # chmod 755 do katalogów:
chmod 755 /home/user oraz # chmod 755 /home/user/public_html
4. By strona uruchamiała się należy dołączyć do apache2 moduł:

a2enmod userdir

5. Każdy użytkownik będzie miał dostęp do strony z poziomu przeglądarki przez adres domeny lub IP. Odwołuje się przez tyldę.

`http://IP/~user` lub <http://domena/~user>

Inne moduły apache2

```
< /etc/apache2
.n
/..
/conf-available
/conf-enabled
/mods-available
/mods-enabled
/sites-available
/sites-enabled
apache2.conf
envvars
magic
ports.conf
```

Moduły dostępne są w katalogu `/etc/apache2/mods-available`

By włączyć moduł podajemy jego nazwę w komendzie systemowej `# a2enmod`.

Przykład: moduł `dir.conf` odpowiada za obsługę plików stron www. Można dodać tu nazwę pliku strony np. `strona.html`. Będzie ona domyślną wczytywaną z katalogu. Myślę, że kolejność wpisów ma znaczenie i jako pierwszy w katalogu zostanie wczytany `index.html`.

```
/etc/apache2/mods-available/dir.conf [---] 0 L:[ 1+ 0 1/ 2] *(0 / 77
DirectoryIndex index.html index.cgi index.pl index.php index.xhtml index.htm
```

Wydanie tu polecenia `# a2enmod dir` powoduje wyświetlenie informacji, że moduł jest włączony. Po dokonaniu wpisu do modułu należy zrestartować `apache2`.

```
root@server:/var/www/domena.org# a2enmod dir
Module dir already enabled
root@server:/var/www/domena.org#
```

Strona https

By uruchomić stronę z obsługą SSL edytujemy w pico wzorcowy plik: default-ssl.conf.

1. Zapisujemy go pod nazwą domeny z ssl. Robimy to przez pico. Pico (ctrl+o, zmiana nazwy, enter, potwierdzenie yes)
2. Wprowadzamy do pliku potrzebne dane [katalog strony, nazwę domeny, alias itd.]
3. Plik *.conf powinien mieć odpowiedni katalog strony, która ma uruchamiać się przez ssl.
4. Włączamy stronę przez: **# a2ensite strona.org.conf**
5. Restartujemy apache2
6. Włączamy moduł: ssl wydając polecenie: **# a2enmod ssl**
7. Strona uruchamia się przez https, ale zwraca problemy zabezpieczeń.

Hasło do katalogu ze stroną

Każdy katalog ze stroną www możemy zabezpieczyć hasłem.

1. W katalogu wirtualnym strony między <VirtualHost> wstawiamy <Directory>

```
<VirtualHost *:80>
ServerName zse2.pl
ServerAlias www.zse2.pl
ServerAdmin admin@zse2.pl
DocumentRoot /var/www/zse2.pl

<Directory /var/www/zse2.pl>
AllowOverride AuthConfig
</Directory>_

</VirtualHost>
```

Określamy w ten sposób hasłowany katalog.

2. W katalogu strony /var/www/strona tworzymy plik ukryty z kropką z przodu: .htaccess

```
< /var/www/zse2.pl
.n      Name
/..
.htaccess
.htpasswd
index.html
```

3. W pliku wpisujemy:

AuthName „Informacja – dowolna o autoryzacji, hasłach itd.”

```
/var/www/zse2.pl/.htaccess  [----] 18 L:[
AuthName "Autoryzacja"
AuthUserFile /var/www/zse2.pl/.htpasswd
AuthType basic
Require valid-user_
```

4. W katalogu strony wydajemy polecenie tworzące hasło i login oraz plik ukryty: .htpasswd

- Polecenie: **htpasswd -c .htpasswd user**

Gdzie user to nazwa użytkownika. Apache zapyta nas o hasło do konta.

5. Wystarczy teraz zresetować Apache2 i nasza strona powinna pytać o login oraz hasło.

- Jest to dość częsty sposób zabezpieczenia strony www. Można przeczytać o tym w manualu: `man htpasswd`, zobaczyć plik: `apache2.conf`.

- Pewne pliki: `.htpasswd` oraz `.htaccess` są używane od dawna w systemie Linux/UNIX.

Monitorowanie serwera Apache2:

- **Netdata** - to narzędzie do monitorowania w czasie rzeczywistym, które pozwala uzyskać pełny obraz wydajności systemu. Można go używać do śledzenia zasobów, takich jak CPU, pamięć i dyski, oraz do monitorowania serwera Apache.

apt install netdata

Port, na którym działa Netdata:

- Domyślnie Netdata działa na porcie 19999. Możesz zmienić port, edytując linię:

[web]

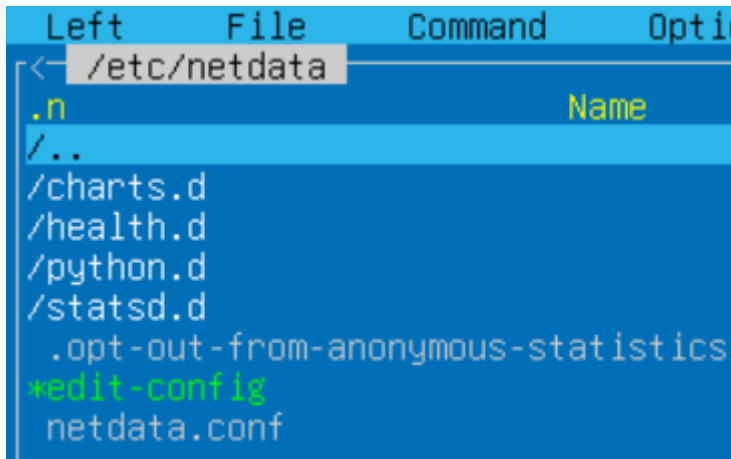
bind to = 127.0.0.1:19999

Dostęp zdalny:

- Aby zezwolić na dostęp do Netdata z zewnątrz, zmień wartość `bind to` na `0.0.0.0` w sekcji `[web]`:

`[web]`

`bind to = 0.0.0.0:19999`



```
Left  File  Command  Optio
<- /etc/netdata
.n Name
/..
/charts.d
/health.d
/python.d
/statsd.d
.opt-out-from-anonymous-statistics
+edit-config
netdata.conf
```

Integracja z innymi systemami

Netdata wspiera integrację z innymi systemami monitorującymi, takimi jak **Prometheus**, **Grafana** i inne. Możesz skonfigurować Netdata, aby eksportował dane do tych systemów, co pozwala na bardziej zaawansowane analizy i wizualizacje.

Aby włączyć eksport do **Prometheus**, wystarczy edytować plik konfiguracyjny:

```
sudo nano /etc/netdata/netdata.conf
```

I włączyć eksport do Prometheus:

[backend]

```
enabled = yes
data source = statsd
type = prometheus
destination = localhost:9090
```

```

/etc/netdata/netdata.conf [----] 28 L:[ 1+25 26/ 26] *(551 / 551b)
# NetData Configuration
# The current full configuration can be retrieved from the running
# server at the URL

# http://localhost:19999/netdata.conf

[global]
<----->run as user = netdata
<----->web files owner = root
<----->web files group = root
<-----># Netdata is not designed to be exposed to potentially hostile
<-----># networks. See https://github.com/netdata/netdata/issues/164
<----->bind socket to IP = 127.0.0.1

[web]
bind to = 0.0.0.0:19999

[plugins]
apache2 = yes
mysql = yes

[backend]
enabled = yes
data source = statsd
type = prometheus
destination = localhost:9090_

```

```

root@server:/var/www/domena.org# systemctl restart netdata
root@server:/var/www/domena.org# _

```

- **Webalizer** to popularne narzędzie do analizy logów serwera WWW, w tym Apache2. Jest to aplikacja służąca do generowania raportów na podstawie danych zapisanych w logach serwera, takich jak dostęp do stron internetowych, liczba wizyt, popularność poszczególnych zasobów czy źródła ruchu. Webalizer oferuje łatwą w użyciu wizualizację danych w formie raportów HTML.

```
# sudo apt-get update
```

```
# sudo apt-get install webalizer
```

Po instalacji warto zmienić plik logów w konfiguracji webalizera.

```

<-- /etc/webalizer
.n                                     Name
/..
webalizer.conf
webalizer.conf.sample

LogFile /var/log/apache2/access.log

```

Po wygenerowaniu raportów Webalizer zapisuje je w formie plików HTML, które możesz przeglądać w dowolnej przeglądarce internetowej. Jeśli raporty zostały zapisane w katalogu `/var/www/webalizer/` to warto przygotować poprawny host wirtualny.

Uruchamiamy webalizer:

webalizer

```
root@server:/var/www/domena.org# webalizer
Webalizer V2.23-08 (Linux 6.8.0-41-generic x86_64) locale: /var/www/webalizer
Using logfile /var/log/apache2/access.log (clf)
Creating output in /var/www/webalizer
Hostname for reports is 'server'
History file not found...
Previous run data not found...
Saving current run data... [03/05/2025 07:04:51]
Generating report for March 2025
Saving history information...
Generating summary report
9 records in 1 seconds, 9/sec
root@server:/var/www/domena.org#
```

http://twoja_domena/webalizer/