

Ćwiczenia 16 instalacja i konfiguracja serwera Apache



1. Na stacji otwórz stronę: httpd.apache.org/docs/
2. Zaloguj się na swoje konto.
3. Wyдай komendę: `sudo apt purge apache2`
4. Zainstaluj pakiety: `sudo apt install apache2 openssl libssl-dev links lynx -y`
5. Sprawdź status serwer komendą: `systemctl status apache2`
6. Sprawdź poprawność konfiguracji: `sudo apache2ctl configtest`
7. Sprawdź na pierwszym terminalu logi: `journalctl -f`
8. Sprawdź czy istnieje proces dla serwera komendą: `ps aux | grep apache`
 - a) Uruchomić przeglądarkę i sprawdzić na 3 sposoby działanie wpisując:

`lynx localhost | links 127.0.0.1 | lynx ip serwera`

(pomoc: `ip addr add 10.11.12.13/24 dev enp3s0 | ip link set enp3s0 up`)
9. Analogicznie przetestuj serwer linkowy ze stacji, jeśli nie działa dostosuj zaporę, należy otworzyć port **80** lub dodać usługę: `sudo ufw allow 'Apache Full'`

```

andrzej@servubu:~$ sudo ufw allow 'Apache Full'
Rules updated
Rules updated (v6)
andrzej@servubu:~$ sudo ufw status
Status: inactive
andrzej@servubu:~$ sudo ufw enable
Firewall is active and enabled on system startup
andrzej@servubu:~$ sudo ufw status
Status: active

```

To	Action	From
--	-----	----
Apache Full	ALLOW	Anywhere
Apache Full (v6)	ALLOW	Anywhere (v6)

```

andrzej@servubu:~$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

```

To	Action	From
--	-----	----
80,443/tcp (Apache Full)	ALLOW IN	Anywhere
80,443/tcp (Apache Full (v6))	ALLOW IN	Anywhere (v6)
10. Sprawdź połączenie z pomocą **wireshark**. (filtruj ruch po http)
11. Popraw wygląd swojej strony. (Stwórz plik: `/var/www/html/index.html`) Sprawdź w przeglądarce.

12. Dodać możliwość tworzenia stron www przez użytkowników systemowych: np.

http://localhost/~twoje_konto (wskazówki: utwórz katalog public_html w swoim katalogu domowym)

```

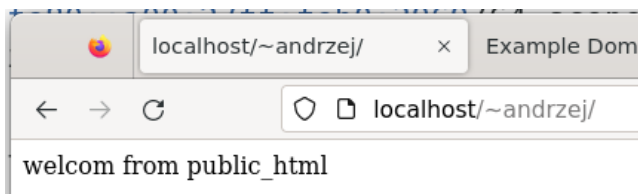
andrzej@servubu:~$ cat /etc/apache2/mods-available/userdir.conf
<IfModule mod_userdir.c>
    UserDir public_html
    UserDir disabled root

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

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
andrzej@servubu:~$ sudo a2enmod userdir
Enabling module userdir.
To activate the new configuration, you need to run:
    systemctl restart apache2
andrzej@servubu:~$ sudo systemctl restart apache2

```

13. Przetestuj stronę (lynx localhost/~twoje_konto):



14. Zmodyfikuj następujące parametry pracy serwera, za każdym razem sprawdzamy działanie w przeglądarce:

a) Nasłuchiwanie na porcie 81 (/etc/apache2/ports.conf),

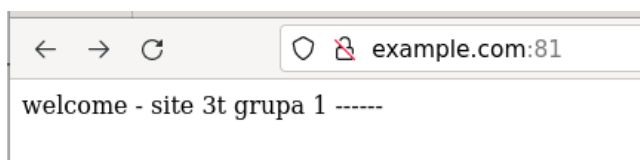


b) Ustaw ServerName www.example.com:81 (/etc/apache2/sites-available/000-default.conf)

```

andrzej@servubu:/etc/apache2/sites-available$ sudo vi /etc/hosts
andrzej@servubu:/etc/apache2/sites-available$ sudo cat /etc/hosts
127.0.0.1 localhost
127.0.1.1 servubu
10.11.12.13 example.com example

```



c) Plik strony w lokalizacji /var/www/twoje_konto/html/index.html (zawartość strony nowa)
wskazówka zmień DocumentRoot.

```

andrzej@servubu:/etc/apache2/sites-available$ sudo cp 000-default.conf nowa.conf

```

```
/etc/apache2/sites-available$ sudo vi nowa.conf
/etc/apache2/sites-available$ sudo a2ensite nowa.conf
```

wa.

new configuration, you need to run:

ad apache2

```
/etc/apache2/sites-available$ sudo systemctl reload apache2
```

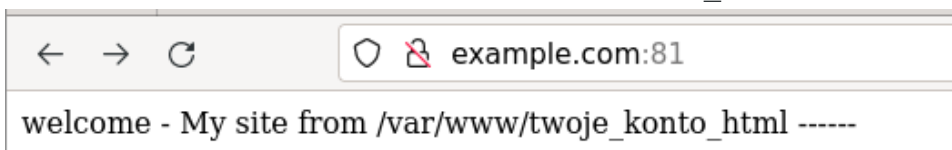
Pamiętaj o

bezpieczeństwie, o skopiowaniu sekcji (shift + insert kopiowanie pomiędzy terminalami)

```
<Directory „/var/www/html”> ... </Directory>
```

```
ServerName example.com
ServerAdmin webmaster@localhost
DocumentRoot /var/www/andrzej/html
<Directory /var/www/andrzej>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>
```

```
andrzej@servubu:/etc/apache2/sites-available$ sudo systemctl reload apache2
andrzej@servubu:/etc/apache2/sites-available$ sudo a2dissite 000-default.conf
Site 000-default disabled.
To activate the new configuration, you need to run:
    systemctl reload apache2
andrzej@servubu:/etc/apache2/sites-available$ sudo a2ensite nowa.conf
Site nowa already enabled
andrzej@servubu:/etc/apache2/sites-available$ sudo systemctl reload apache2
```



d) Zmień wpis dla administratora strony www

```
ServerName example.com
ServerAdmin twoje_imie@localhost
```

e) Zezwól na czytanie poza index.html na inne dokumenty: index.php **egzamin.html i egz.php** (pamiętaj, aby utworzyć te pliki) (podpowiedź: https://httpd.apache.org/docs/2.4/mod/mod_dir.html#directoryindex

```
andrzej@servubu:/etc/apache2$ cat mods-available/dir.conf
<IfModule mod_dir.c>
    DirectoryIndex egzamin.html egz.php index.html index.cgi
    php index.xhtml index.htm
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
andrzej@servubu:/etc/apache2$ sudo a2enmod dir
Module dir already enabled
```

```
:/var/www/andrzej/html$ sudo mv index.html egzamin.html
```

- f) Zmień poziom logów z warn na info lub debug (/etc/apache2/apache2.conf),

```
#LogLevel warn
LogLevel info
```

- g) Zmień domyślny content z UTF-8 na ISO-8859-1

```
andrzej@servubu:/etc/apache2$ cat conf-available/charset.conf
# Read the documentation before enabling AddDefaultCharset.
# In general, it is only a good idea if you know that all your files
# have this encoding. It will override any encoding given in the files
# in meta http-equiv or xml encoding tags.

#AddDefaultCharset UTF-8
AddDefaultCharset ISO-8859-1

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
andrzej@servubu:/etc/apache2$ sudo a2enconf charset
Conf charset already enabled
andrzej@servubu:/etc/apache2$ sudo systemctl reload apache2
```

- h) Zmodyfikuj komunika 404 (wsk. ErrorDocument 404)

- i) Utwórz 2 serwery wirtualne (skopiuj plik 000-default.conf na /etc/apache2/sites-available/www1-example-com.conf,

pamiętaj o stworzeniu plików index.html i przeładowaniu serwera: *sudo systemctl reload apache2*

pomoc: <http://httpd.apache.org/docs/2.4/vhosts/>):

```
andrzej@servubu:/etc/apache2/sites-available$ sudo cp 000-default.conf www1-example-com.conf
andrzej@servubu:/etc/apache2/sites-available$ sudo cp 000-default.conf www2-example-org.conf
andrzej@servubu:/etc/apache2/sites-available$ sudo a2ensite www1-example-com
Enabling site www1-example-com.
To activate the new configuration, you need to run:
systemctl reload apache2
andrzej@servubu:/etc/apache2/sites-available$ sudo a2ensite www2-example-org
Enabling site www2-example-org.
To activate the new configuration, you need to run:
systemctl reload apache2
andrzej@servubu:/etc/apache2/sites-available$ sudo systemctl reload apache2
```

```
:/var/www$ sudo mkdir www1-example-com
:/var/www$ sudo mkdir www2-example-org
:/var/www$ sudo vi www1-example-com/index.html
:/var/www$ sudo vi www2-example-org/index.html
```

```
<VirtualHost 10.11.12.13:81>
```

```
    ServerName www1.example.com
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/www1-example-com
    Protocols h2 http/1.1
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/www1.example.com/error.log
    CustomLog ${APACHE_LOG_DIR}/www1.example.com/access.log combined
```

```
</VirtualHost>
```



10.11.12.13:81

welcome site vhost 1

```
andrzej@servubu:/etc/apache2/sites-available$ sudo a2enmod ht
Enabling module http2.
```

To activate the new configuration, you need to run:
systemctl restart apache2

```
root@:/etc/www2-example-org.com
```

```
<VirtualHost 172.20.30.50:81>
```

```
    ServerName www2.example.org
    ServerAdmin webmaster@www2.example.org
    DocumentRoot /var/www/www2-example-org
    ErrorLog ${APACHE_LOG_DIR}/www2-example-org/error.log
    CustomLog ${APACHE_LOG_DIR}/www2-example-org/access.log combined
```

```
</VirtualHost>
```



www2.example.org:81

welcome site vhost 2

j) Sprawdź stronę poleceniem curl. np. curl http://10.11.12.13:81 -sSI

```

andrzej@servubu:/var/www$ curl -sSI http://www1.example.com:81
HTTP/1.1 200 OK
Date: Mon, 02 Jan 2023 15:29:32 GMT
Server: Apache/2.4.41 (Ubuntu)
Upgrade: h2
Connection: Upgrade
Last-Modified: Mon, 02 Jan 2023 15:05:46 GMT
ETag: "15-5f1494934e300"
Accept-Ranges: bytes
Content-Length: 21
Content-Type: text/html; charset=ISO-8859-1

```

k) Sprawdź konfigurację serwera poleceniem: *sudo apache2ctl -S*

```

andrzej@servubu:/var/www$ sudo apache2ctl -S
VirtualHost configuration:
10.11.12.13:81      www1.example.com (/etc/apache2/sites-enabled/www1-example-com.conf:1)
172.20.30.50:81    www2.example.org (/etc/apache2/sites-enabled/www2-example-org.conf:1)
*:81              example.com (/etc/apache2/sites-enabled/nowa.conf:1)
ServerRoot: "/etc/apache2"
Main DocumentRoot: "/var/www/html"
Main ErrorLog: "/var/log/apache2/error.log"

```

l) Dodaj jeszcze dwa serwery wirtualne, ale oparte o nazwy, wykorzystaj poniższą odpowiedź:

```
<VirtualHost 203.0.113.1:82>
```

```

    ServerName www.example.net
    ServerAdmin webmaster@example.net
    DocumentRoot /home/andrzej/example.net
    ServerAlias example.net

    <Directory /home/andrzej>
        Options Indexes FollowSymLinks
        AllowOverride None
        Require all granted
    </Directory>

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

```

```
</VirtualHost>
```



```
net test 3
```

```
<VirtualHost *:83>
```



```
ServerName other.example.com

DocumentRoot "/www/ other.example.com "

</VirtualHost>
```

Przywróć nasłuchiwanie serwera na port 81!!!

Druga część dla połączeń szyfrowanych:

15. Sprawdź czy istnieją certyfikaty dla serwera:

```
andrzej@servubu:~$ zcat /usr/share/doc/apache2/README.Debian.gz | less
```

16. Włącz obsługę ssl: **sudo a2enmod ssl**

```
andrzej@servubu:~$ sudo a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create s
elf-signed certificates.
To activate the new configuration, you need to run:
    systemctl restart apache2
```

```
andrzej@servubu:~$ sudo systemctl restart apache2
```

17. Uruchomić przeglądarkę i sprawdzić na 3 sposoby działanie wpisując **https://localhost -k | 127.0.0.1 | ip** serwera. Sprawdź też **localhost:443**



welcome - site 3t grupa 1 -----

```
lynx https://localhost
```

18. Jeżeli są problemy z uruchomieniem stron to:

```
andrzej@servubu:/var/log/apache2$ sudo a2ensite default-ssl
Enabling site default-ssl.
```

```
To activate the new configuration, you need to run:
    systemctl reload apache2
```

```
andrzej@servubu:/var/log/apache2$ sudo systemctl reload apache2
```

19. Sprawdź aktywne połączenia ze swoim serwerem komendą: **netstat | grep lub ss -l | grep**

20. Analogicznie przetestuj serwer apache ze stacji, jeśli nie działa dostosuj zaporę, należy otworzyć port **443** lub dodać usługę)

21. Sprawdź połączenie z pomocą **wireshark**. (filtruj ruch po https)

22. Sprawdź zawartość logów.

23. Dodać możliwość tworzenia stron www przez użytkowników systemowych: np.

https://localhost/~twoje_konto (wskazówka: public_html)

24. Utwórz serwer wirtualny, który:

- a) Działa na ip 10.11.12.13 i porcie 443
- b) Pliki stron znajdują się w lokalizacji /var/www/ssl/twoje_konto
- c) Ustaw obsługę protokołu **HTTP/2** (wsk. <http://httpd.apache.org/docs/2.4/howto/http2.html>)

25. Przykładowa realizacja:

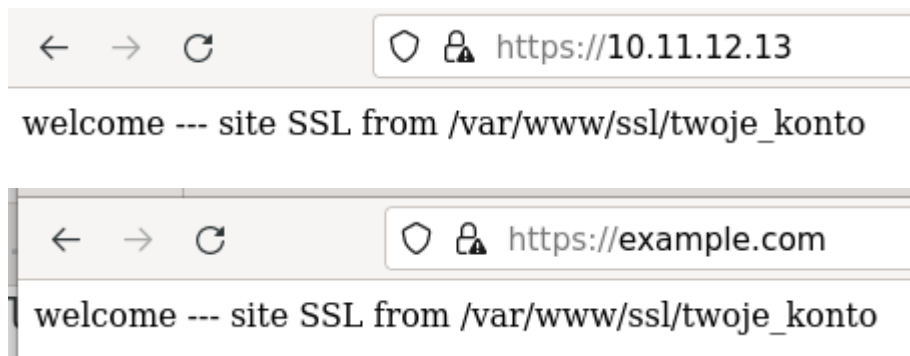
```
<IfModule mod_ssl.c>
  <VirtualHost 10.11.12.13:443>
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/ssl/andrzej
    Protocols h2 h2c http/1.1
    LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
    SSLEngine on
    SSLCertificateFile      /etc/ssl/certs/ssl-cert-snakeoil.pem
    SSLCertificateKeyFile   /etc/ssl/private/ssl-cert-snakeoil.key

    <FilesMatch "\.(cgi|shtml|phtml|php)$">
      SSLOptions +StdEnvVars
    </FilesMatch>
    <Directory /usr/lib/cgi-bin>
      SSLOptions +StdEnvVars
    </Directory>

  </VirtualHost>
</IfModule>
```

26. Przetestuj działanie w przeglądarce:



27. Utwórz **drugi** serwer wirtualny, który:

- a) Działa na ip zgodnie z RFC 5737 - IPv4 Address Blocks Reserved for Documentation (<https://tools.ietf.org/html/rfc5737>) i porcie 443
Wskazówka: możesz skorzystać z 198.51.100.0/24 (TEST-NET-2)
- b) Pliki stron znajdują się w lokalizacji /var/www/ssl/2/twoje_konto
- c) Nazwa strony: example.net
- d) Ustaw poziom logów na notice lub crit

28. Sprawdź oba serwery wirtualne.

29. Dla strony: <https://198.51.100.1/> użyj sprawdzenia w chrome->zbadaj->Lighthouse->raport

30. Sprawdź stronę <https://10.11.12.13> za pomocą curl czy obsługuje HTTP2.

Przykład curl -I --http2 https://google.pl

31. Dla wirtualnych hostów pracujących na porcie 81 wykonaj przekierowanie ruchu do https.
32. Dodatkowe zadanie: dla ip 192.0.2.1 i portu 443 (RFC 5737 192.0.2.0/24 (TEST-NET-1)) wygeneruj własny certyfikat w oparciu o materiały z wykładu.
33. Zastosuj ServerAlias <http://httpd.apache.org/docs/2.4/mod/core.html#serveralias> dla nowego wirtual hosta.
34. Wykonaj kopię edytowanych plików.
35. KONIEC