

Wprowadzanie

Apache jest to otwarty i darmowy (open-source) serwer http dostępny dla wielu systemów operacyjnych.

Serwer http, inaczej **serwer WWW**, jest to program komputerowy działający na serwerze, obsługujący żądania protokołu komunikacyjnego http. Z serwerem WWW łączy się przeglądarka internetowa będąca jego klientem, aby pobrać i wyświetlić użytkownikowi stronę internetową.

przeglądarka internetowa – serwer WWW

(klient – serwer)

Protokół http działa na porcie 80/tcp, natomiast https (szyfrowana wersja protokołu http) na porcie 443/tcp.

Serwer WWW może współgrać z usługami innego, równoległe działającego oprogramowania, np. MySQL, PHP.

Bardzo popularnym rozwiązaniem jest LAMP = Linux + Apache + MySQL + PHP (system operacyjny, serwer WWW, serwer baz danych, język kodowania stron internetowych). Wszystko razem połączone tworzy jedność umożliwiającą obsługę i wyświetlanie nawet rozbudowanych serwisów internetowych.

Apache w systemie Linux

Oprogramowanie serwera Apache jest natywnie dostępne w systemowych repozytoriach wszystkich dystrybucji pod nazwą apache2.

```
apt update
```

```
apt install apache2
```

Po instalacji pakietu utworzona zostaje usługa o takiej samej nazwie. Zarządzanie nią:

```
systemctl start/stop/restart/enable/disable/reload apache2
```

Gdy zainstalujemy i uruchomimy usługę możemy sprawdzić jej działanie. W tym celu wystarczy uruchomić dowolną przeglądarkę internetową i wpisać `http://adres_ip_serwera` (czyli np. `http://localhost`, jeżeli przeglądarkę uruchomiliśmy na tym samym hoście co uruchomiony jest serwer, albo `http://1.1.1.1`, jeżeli serwer jest uruchomiony na innym hoście, wstawiając oczywiście odpowiedni adres IP).

Dobłą praktyką jest zmiana właściciela katalogu `/var/www/html` (z parametrem `-R`, aby zmiany dotyczyły również wszystkich podkatalogów) na `www-data:www-data`.

Pliki konfiguracyjne serwera

Pliki konfiguracyjne serwera Apache znajdują się w katalogu [/etc/apache2](#).

Głównym plikiem jest [apache2.conf](#), w którym oprócz opcji konfiguracyjnych znajdują się również odnośniki do innych plików konfiguracyjnych.

W pliku [envvars](#) znajdują się stałe wartości tekstowe używane w plikach konfiguracyjnych. Na przykład domyślny użytkownik i grupa, które są tworzone w czasie instalacji serwera Apache nazywają się `www-data`. Serwer działa w oparciu o uprawnienia tego użytkownika i tej grupy. Również pliki i katalogi udostępniane przez serwer WWW powinny mieć tak ustawione uprawnienia, aby były widoczne dla użytkownika `www-data`. Wartość `www-data` jest zapisana w pliku `envvars` pod stałą `APACHE_RUN_USER`, dlatego w innych plikach konfiguracyjnych możemy się odwoływać do tej wartości właśnie poprzez nazwę `APACHE_RUN_USER`.

Plik [ports.conf](#) pozwala zmienić domyślne porty połączenia ze stronami internetowymi. Dla połączeń nieszyfrowanych po protokole `http` jest to port 80, natomiast połączenia szyfrowane po protokole `https` odbywają się domyślnie na porcie 443.

Główny plik konfiguracyjny serwera Apache jest zazwyczaj uzupełniany przez pomocnicze pliki konfiguracyjne. Znajdują się one w katalogach [mods-available](#), [conf-available](#) i [sites-available](#) jako osobne pliki konfiguracyjne do witryn obsługiwanych przez nasz serwer. Jest to dobre rozwiązanie, ponieważ na serwerze może być hostowane wiele witryn internetowych. Wtedy każda z nich będzie miała dedykowane pliki konfiguracyjne.

W [mods-available](#) znajdują się moduły dołączane do witryny, np. `auth_basic` zapewnia prostą autoryzację użytkownika na stronie.

W katalogu [conf-available](#) znajdziemy bardziej szczegółowe pliki konfiguracyjne, typu komunikaty obsługi podstawowych błędów HTTP w językach innych niż angielski.

Z kolei w [sites-available](#) znajdują się pliki odpowiedzialne za tworzenie wirtualnych hostów, czyli pozwalające umieszczać kilka stron, widocznych pod różnymi adresami, ale znajdujących się fizycznie na jednym serwerze.

Pliki z katalogów z nazwą kończącą się na *-available* są plikami możliwymi do użycia. O tym, które z nich zostaną dołączone i aktywowane na naszym serwerze, decydują dowiązania symboliczne znajdujące się w katalogach z zakończeniem *-enabled*. Są to katalogi: *mods-enabled*, *conf-enabled* i *sites-enabled*.

We wszystkich plikach konfiguracyjnych serwera Apache znak # oznacza początek linii komentarza. Należy również pamiętać, że w systemach z rodziny Linux małe i duże litery to różne znaki.

Położenie głównego katalogu zawierającego pliki stron jest określone w plikach konfiguracyjnych przez zmienną *DocumentRoot*. Najczęściej jest to ścieżka */var/www/html*.

Narzędzia serwera Apache

Apache natywnie udostępnia narzędzia ułatwiające pracę. Dzięki nim możemy zarządzać modułami, konfiguracjami oraz wirtualnymi hostami.

Zacznijmy od modułów. Dostępne moduły znajdziemy w katalogu *mods-available*. Aby je włączyć musimy wykorzystać ***a2enmod*** (enable module = włącz moduł) i nazwa modułu jaki nas interesuje. W tym momencie włączony moduł powinien się pojawić w katalogu *mods-enabled*. Do wyłączenia służy narzędzie ***a2dismod*** (disable module = wyłącz moduł).

Analogicznie sytuacja dotyczy zaawansowanych konfiguracji. Dostępne są one w katalogu *conf-available*, a po włączeniu *conf-enabled*. Narzędzia obsługujące to ***a2enconf*** oraz ***a2disconf***.

Trzecią grupą są wirtualne hosty. Tworzymy je w katalogu *sites-available*. Jeżeli chcemy „ukazać światu” daną witrynę włączamy ją poprzez ***a2ensite***. Jeżeli nie chcemy już jej więcej serwować, ale chcemy zostawić plik konfiguracyjny wyłączamy ją z wykorzystaniem ***a2dissite***.

Na przykład:

```
a2ensite firmaa.pl.conf
```

```
a2dissite firmaa.pl.conf
```

Jak widzimy podajemy dokładną nazwę pliku, który zawiera konfigurację dla danej witryny. Po użyciu narzędzia otrzymamy informację o konieczności przeładowania usługi (`systemctl reload apache2`) co oczywiście musimy wykonać.

Zmiana portu na którym nasłuchuje serwer

Konfiguracja portów serwera znajduje się w pliku `ports.conf` oraz domyślnym pliku wewnątrz katalogu `sites-enabled`.

W pierwszym z w/w plików zmieniamy numer w dyrektywie `Listen 80`, natomiast w drugim z plików edytujemy linię `<VirtualHost *:80>`.

Aby zmiany weszły w życie należy ponownie uruchomić serwer WWW. Dostęp do strony odbywa się wówczas przez dodanie portu do adresu IP/nazwy domenowej (`http://1.1.1.1:8888`).

Wirtualne hosty

Dzięki wirtualnym hostom możemy uruchomić na jednym serwerze WWW wiele witryn internetowych. W ten sposób na naszym serwerze możemy mieć jednocześnie stronę dla FirmyA (`http://firmaa.pl`), FirmyB (`http://firmab.pl`) oraz FirmyZ (`http://firmaz.pl`).

Dzięki wirtualnym hostom możemy wprowadzać dowolne konfiguracje dla każdej z witryn. Możemy na przykład:

- określić katalog główny witryny,
- stworzyć oddzielną politykę bezpieczeństwa,
- używać różnych certyfikatów SSL (każda strona będzie posiadała swoje).

Wirtualne hosty posiadają dwie możliwości, mogą być oparte na adresach IP (wówczas serwer działa na różnych adresach IP, każda strona to inny adres IP), bądź nazwach domenowych (serwer działa na różnych nazwach domenowych, każda strona to inna domena). Skupimy się jednak na tej drugiej opcji.

Od strony DNS: jest jeden serwer WWW i kilka domen. Wszystkie domeny wskazują na jeden, ten sam adres IP serwera WWW (rekord A). Powyższy przykład będzie wyglądał:

Serwer WWW: 1.1.1.1

www.firmaa.pl -> serwer WWW znajduje się pod adresem 1.1.1.1

www.firmab.pl -> serwer WWW znajduje się pod adresem 1.1.1.1

www.firmaz.pl -> serwer WWW znajduje się pod adresem 1.1.1.1

Dobłą praktyką jest wówczas utworzenie wewnątrz katalogu `/var/www/html` oddzielnych podkatalogów dla każdej z witryn, nazywając je nazwami domen (czyli np. *firmaa.pl*)

Konfiguracja wirtualnych hostów

Do debugowania konfiguracji wirtualnych hostów wykorzystujemy `apachectl -S`.

Zacznijmy od stworzenia prostych wirtualnych hostów. Każdy host to oddzielny plik, zawierający w nazwie domenę i rozszerzenie *conf* (czyli np. *firmaa.pl.conf*). Tworzymy je w katalogu *sites-available*. Po utworzeniu musimy je aktywować odpowiednim narzędziem opisanym powyżej.

```
<VirtualHost *:80>
```

```
    DocumentRoot "/var/www/html/firmaa.pl"
```

```
    ServerName firmaa.pl
```

```
</VirtualHost>
```

```
<VirtualHost *:80>
```

```
    DocumentRoot "/var/www/html/firmab.pl"
```

```
    ServerName firmab.pl
```

```
</VirtualHost>
```

Aby zmiany wprowadzane w plikach konfiguracyjnych wchodziły w życie należy wykonać reload usługi.

Każda strona musi być zdefiniowana wewnątrz znacznika *VirtualHost* (każda ma swój oddzielny). * oznacza domyślny adres na jakim nasłuchuje serwer. W tym miejscu możemy podać konkretny adres IP, ale jeżeli przyjdzie konieczność zmiany adresu serwera będziemy musieli pamiętać, aby zmienić również w tym miejscu. Dyrektywa **DocumentRoot** określa domyślny katalog, gdzie znajdują się pliki danej strony, natomiast **ServerName** jest to nazwa domenowa. W ten sposób po wpisaniu

domeny w przeglądarkę internetową, pakiet zostanie przesłany do serwera, a on wewnątrz wirtualnego hosta odszuka ją i przekieruje do odpowiedniego podkatalogu z plikami witryny.

A co w sytuacji, gdy domena kieruje na ten serwer, ale nie ma stworzonego wirtualnego hosta? Zostanie wykonana domyślna akcja dla serwera (zdefiniowana w pliku *000-default.conf*), więc użytkownik zostanie przekierowany na główny *DocumentRoot*.

Dyrektywa ***ServerAdmin*** definiuje adres mailowy administratora. Na ten adres mogą być wysyłane informacje o błędach na witrynie. Pozwala on również na kontakt użytkowników z webmasterem danej witryny.

Kolejną dyrektywą jest ***ServerAlias***. Definiuje ona aliasy (inne nazwy), które skierują nas na nazwę domenową zdefiniowaną w ***ServerName***. W tym miejscu możemy umieścić np. domenę z przedrostkiem *www* (w tym miejscu warto podkreślić, że dla Apache domena *www.domena.pl* i domena *domena.pl* to dwie różne domeny), bądź inne subdomeny. Możemy tutaj również wpisać całkowicie inną domenę, która będzie kierowała na daną witrynę.

Ważną kwestią w administracji serwerami są logi. Dla przejrzystości warto dla każdego wirtualnego hosta tworzyć oddzielne pliki z logami. Apache rozdziela logi z błędami od logów dostępowych. Dyrektywy odpowiedzialne za definicję plików to:

ErrorLog *\${APACHE_LOG_DIR}/error_domena.log*

CustomLog *\${APACHE_LOG_DIR}/access_domena.log combined*

Pierwsza z nich określa lokalizację pliku z błędami, natomiast druga zawiera logi dostępowe. Po pliku mamy określenie *combined*, definiuje ono format logów.

Format *common*:

```
127.0.0.1 - frank [10/Oct/2000:13:55:36 -0700] "GET /apache_pb.gif
HTTP/1.0" 200 2326
```

Format *combined*:

```
127.0.0.1 - frank [10/Oct/2000:13:55:36 -0700] "GET /apache_pb.gif
HTTP/1.0" 200 2326 "http://www.example.com/start.html" "Mozilla/4.08
[en] (Win98; I ;Nav)"
```

Małe wtrącenie, konfigurację wykonaliśmy w oparciu o zmienną `APACHE_LOG_DIR`. Określa ona domyślny katalog logów dla Apache, który został zdefiniowany w pliku `envvars`.

W tym miejscu uruchommy logowanie zdarzeń dla naszych wirtualnych hostów i sprawdźmy jak wyglądają i co zawierają logi serwera Apache. Dostępne one są w lokalizacji `/var/log/apache2/`.

Przejdźmy do konfiguracji zachowania danego katalogu. Aby konfigurować główny katalog dla danego hosta wykorzystamy dyrektywę **<Directory /var/www/html/firmaa.pl>**.

Domyślne zachowanie serwera Apache wygląda następująco:

1. wyświetl plik `index.html`
2. jeżeli nie ma powyższego, wyświetl `index.php`
3. jeżeli nie ma żadnego z tych wylistuj zawartość katalogu (dość niebezpieczna opcja - po co ktoś ma widzieć jakie pliki przechowujemy na serwerze; w niektórych mogą być przechowywane konfigurację połączenia do bazy SQL - dane logowania w formie jawnego tekstu)

Aby zmienić tę kolejność, albo wskazać plik, który ma się domyślnie wyświetlać jako pierwszy wykorzystamy **DirectoryIndex**. Może on wskazywać konkretny plik lub kolejne pliki, które mają być wyświetlane w przypadku braku poprzedniego. Czyli na przykład:

`DirectoryIndex plik.html`

`DirectoryIndex index.php index.html plik.html`

Poszczególne pliki rozdzielamy spacją.

A co w przypadku, gdy nie będzie żadnego z wymienionych tutaj plików? Serwer wylistuje nam wszystkie pliki w katalogu. Jest to dość niebezpieczne i niepożądane. Aby to wyłączyć musimy wykorzystać dyrektywę **Order** i ustawić ją jako `Allow,Deny`. W tym momencie gdy będziemy próbowali się dostać na witrynę otrzymamy błąd 403.

Jeżeli dobrze przyjrzymy się otrzymanej witrynie na dole znajdziemy wersję oprogramowania Apache. Bardzo niebezpieczne (wystarczy wpisać

w Google wersję i otrzymamy listę otwartych furtek, dla hakera idealne rozwiązanie). Wyłączmy wyświetlanie tego.

W tym celu musimy wyedytować główny plik konfiguracyjny serwera, czyli *apache2.conf* i na końcu dopisać **ServerSignature Off**

Uruchomienie obsługi pliku *.htaccess* (opisany poniżej) odbywa się poprzez dodanie dyrektywy **AllowOverride All**

Powyższa konfiguracja wycięta nam również możliwość podglądu plików, które faktycznie są na serwerze. Aby to włączyć wpisujemy dyrektywę **Allow from all**.

Finalnie nasz plik konfiguracyjny wirtualnego hosta wygląda następująco:

```
<VirtualHost *:80>
    DocumentRoot „/var/www/html/firmaa.pl”
    ServerName firmaa.pl
    ServerAdmin webmaster@firmaa.pl
    ServerAlias www.firmaa.pl
    Error Log ${APACHE_LOG_DIR}/error_firmaa.pl.log
    CustomLog ${APACHE_LOG_DIR}/access_firmaa.pl.log combined

    <Directory /var/www/html/firmaa.pl>
        DirectoryIndex index.php index.html plik.html
        AllowOverride All
        Order Allow,Deny
        Allow from all
    </Directory>
</VirtualHost>
```

Plik *.htaccess*

Plik zawierający konfigurację dla poszczególnych wirtualnych hostów. Umieszcawia się go w głównym katalogu dla danej domeny (*DocumentRoot*). Umożliwia konfigurację rozmaitych funkcji, między

innymi przekierowań (w tym wymuszenie obsługi HTTPS, czy przekierowywanie domeny z www na domenę bez www), blokowanie konkretnych adresów IP, zmiana wersji PHP, zabezpieczenie dostępu do strony hasłem.

Zmiany wprowadzana w tym pliku widoczne są natychmiast – nie jest wymagane ponowne uruchomienie serwera.

Przeanalizujemy dwie z możliwości pliku *.htaccess*. Pierwsza z nich to przekierowanie 301. Służy ono do przekierowania użytkownika z jednej domeny na drugą. Przykładem może być witryna *https://wojst.pl/technikum*, która przekierowuje na serwer z plikami edukacyjnymi.

```
RewriteEngine On
```

```
RewriteCond %{HTTP_HOST} ^(www\.)?domena-stara\.pl [NC]
```

```
RewriteRule (.*) http://domena-nowa.pl/$1 [R=301,L]
```

Warto tutaj zaznaczyć, iż do obsługi przekierowań wykorzystywany jest moduł *mod_rewrite*. Moduł, czyli rozszerzenie / dołożenie dodatkowych funkcjonalności do serwera.

Druga z opcji to zabezpieczenie dostępu do strony hasłem. W tym celu oprócz pliku *.htaccess* będziemy potrzebowali również plik *.htpasswd* (passwd, znajome prawda? ;)).

Do pliku *.htaccess* dodajemy:

```
AuthType Basic
```

```
AuthName "Dostęp zablokowany. Wprowadź hasło."
```

```
AuthUserFile /var/www/html/.htpasswd
```

```
Require valid-user
```

AuthType – sposób uwierzytelnienia

AuthName – tekst, który wyświetli się podczas próby dostępu do strony

AuthUserFile – lokalizacja pliku z bazą użytkowników

Require - daj dostęp jedynie zweryfikowanym użytkownikom

Plik *.htpasswd* ma następującą składnię:

nazwa użytkownika : hasło zaszyfrowane z wykorzystaniem MD5/SHA1 (do szyfrowania możemy posłużyć się witryną

<https://www.web2generators.com/apache-tools/htpasswd-generator>)

user1:\$apr1\$clme56rh\$7kbO8h94VA5UWBNpdayu80