
Serwer WWW

Instalacja i konfiguracja serwera WWW w Linux.

Konfigurowanie wirtualnych serwerów WWW w Linux.

Blokowanie przeglądania zawartości katalogu.

Serwer WWW

- Wprowadzenie do serwera WWW
- Instalacja serwera WWW
- Konfiguracja serwera
- Moduły serwera WWW
- Konfiguracja autentykacji
- Konfiguracja PHP
- Serwery wirtualne
- Certyfikaty SSL (HTTPS)

Wprowadzenie do serwera WWW

Apache ma ponad 50 procent udziału w rynku.

Zalety:

jest stabilny

kilka największych stron takich jak amazon czy IBM go używają

całe oprogramowanie i dodatkowe elementy są na licencji open source

pracuje na wielu platformach (Unix, Linux, Windows)

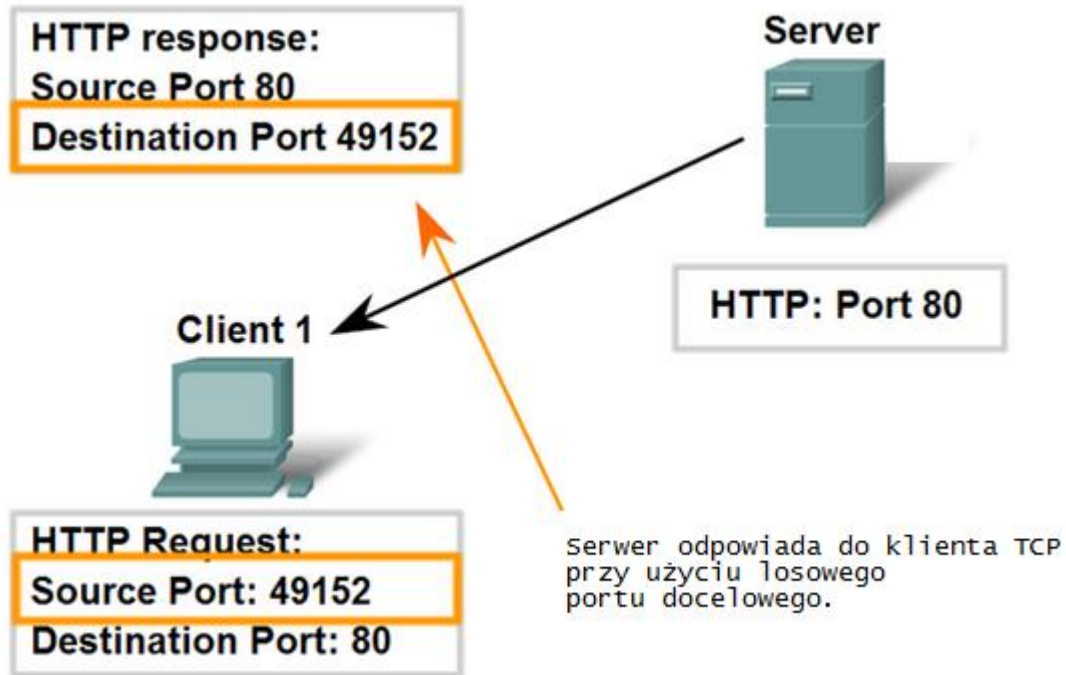
jest elastyczny

jest bezpieczny

Wprowadzenie do serwera WWW

- Apache HTTP Web Server
- Cechy serwera
 - wielowątkowość
 - skalowalność
 - bezpieczeństwo
 - kontrola dostępu/uwierzytelnianie: `mod_authz_host`
- Pakiety `apache2` `apache2-doc` `apache2-utils`
- Demon `/usr/sbin/apache2`
- Skrypt `/etc/init.d/apache2`
- konfiguracja `/etc/apache2/*`, `/var/www/*`
- Porty 80 (http), 443 (https)

Wprowadzenie do serwera WWW



Wprowadzenie do serwera WWW

- Nagłówek klienta

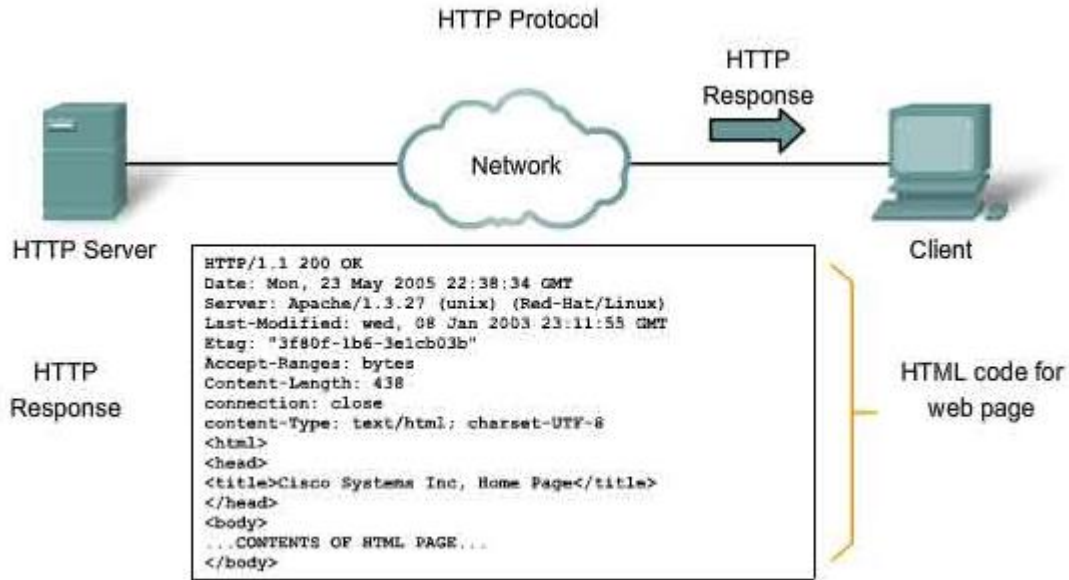
```
GET / HTTP/1.1
Connection: Keep-Alive
User-Agent: Mozilla/5.0 (X11; U; Linux i686)
Host: localhost:80
Accept: text/xml, image/gif, image/jpeg,
image/png...
Accept-Encoding: gzip,deflate
Accept-Language: en-us
Accept-Charset: iso-8859-1,*,utf-8
```

- Nagłówek serwera

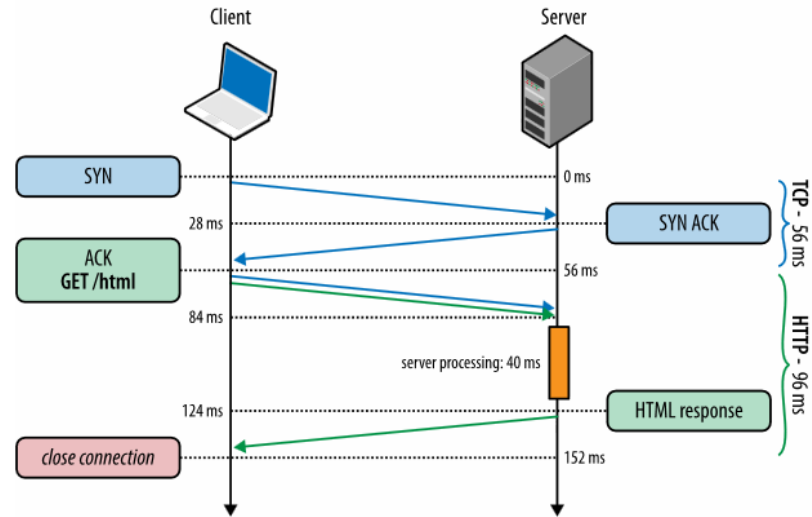
```
HTTP/1.1 200 OK
Date: Thu, 02 Jun 2009 14:03:31 GMT
Server: Apache/2.0.52 (Fedora)
Last-Modified: Thu, 02 Jun 2009 11:41:32 GMT
ETag: "3f04-1f-b80bf300"
Accept-Ranges: bytes
Content-Length: 31
Connection: close
Content-Type: text/html; charset=UTF-8
```

- Pusta linia – nagłówek kompletny
- Transmisja danych

Wprowadzenie do serwera WWW



In response to the request, the HTTP server returns code for a web page.



Instalacja serwera WWW

- Instalacja Apache 2

```
apt-get install apache2 apache2-doc apache2-utils
```

- Instalacja ze źródeł

```
wget http://www.apache.org/dist/httpd/httpd-2.4.23.tar.gz
```

```
gunzip -c httpd-2.4.23.tar.gz |tar xvzf -
```

```
./configure --prefix=/usr/local/httpd
```

```
make
```

```
make install
```


Konfiguracja serwera

Podstawowa konfiguracja w pliku

/etc/apache2/

```
# |-- apache2.conf - podstawowa konfiguracja
# |   |-- ports.conf – konfiguracja portów 80(http) 443(https)
# |   |-- mods-enabled – konfiguracja modułów wywoływanych przez
symlinkki
# |       |-- *.load
# |       |-- *.conf
# |   |-- conf-enabled
# |       |-- *.conf
# |   |-- sites-enabled
# |       |-- *.conf
```

Konfiguracja serwera

- W dystrybucji Debian GNU / Linux zmieniono nazwę oprogramowania z httpd na apache2.
- W systemie Debian znajdziesz plik konfiguracyjny o nazwie apache2.conf w katalogu o nazwie /etc/apache2
- W systemie Ubuntu/Debian Apache przetwarza wszystkie pliki w /etc/apache2/sites-enabled/ (które powinny być dowiązaniem symbolicznymi do plików w katalogach sites-available, zarządzanych przez programy a2ensite i a2dissite)

Konfiguracja serwera

- Zatrzymanie i uruchomienie serwera
/etc/init.d/apache2 (start/stop/reload)
apache2ctl
- Strona domowa
/var/www/

Konfiguracja serwera

- Podstawowa konfiguracja w pliku
 - /etc/apache2/apache2.conf
 - parametry web serwera
 - virtual hosts
 - definicje dostępu
 - mime-types
 - Konfiguracja modułów
- /etc/apache2/mods-enabled*
- /etc/apache2/sites-available*
- włączenie strony *a2ensite*
 - wyłączenie strony *a2dissite*

Konfiguracja serwera

- Opcje konfiguracyjne
 - Server Root *directory-path* – katalog konfiguracyjny `/etc/apache2`
 - Listen [*IP-address:*] *portnumber* – porty
 - ServerName *fully-qualified-domain-name*[: *port*] – identyfikacja samego siebie
 - ServerAdmin *e-mail_address* – mail
 - MaxClient – maksymalna ilość jednoczesnych połączeń
 - LoadModule – używane do dodawania i korzystania z dodatkowych modułów
 - User – określa ID użytkownika z którego uprawnieniami będzie pracował web serwer
 - Group – określa grupę dla procesu Apache
 - Alias – dyrektywa pozwalająca korzystać z zawartości dokumentów trzymanyh w innych lokalizacjach niż DocumentRoot
 - ScriptAlias – określa docelowy katalog skryptów CGI

Moduły serwera WWW

- Większość konfiguracji plik `apache2.conf`
- Inne dyrektywy zawarte w modułach
- Ilość modułów zależna od wersji
- Kompilacja wszystkich modułów
`./configure --enable-module=all`
- Instalacja modułów
`apt-get install [module-name]`

Moduły serwera WWW

mod_cgi – pozwala na wykonywanie skryptów CGI na web serwerze

mod_perl – włącza interpreter Perl w web serwerze

mod_aspdotnet – dostarcza interfejs ASP.NET dla hostów Microsoft obsługujących silnik (engine) ASP.NET

mod_auth_ldap – pozwala na autentykację do serwera Apache na podstawie LDAP

mod_ssl - pozwala na szyfrowanie za pomocą SSL oraz TLS

Konfiguracja autentykacji

- Start z uprawnieniami root
- Praca na koncie innego użytkownika
- Kontrola na poziomie plików i katalogów
- Dostęp kontrolowany opcje allow, order lub deny
 - allow,deny – pozwala na dostęp określonym klientom wszystkim innym zabrania
 - deny, allow – zabrania na dostęp określonym klientom i zezwala wszystkim innym.
- .htaccess

Konfiguracja autentykacji

- `.htaccess`
 - uwzględniony gdy istnieje wpis `AllowOverride` w `Directory`:

```
<Directory />  
Options None  
AllowOverride None  
</Directory >
```
- `AllowOverride` może mieć następujące opcje:
 - `AuthConfig` – można zmieniać opcje dotyczące autoryzacji dostępu do katalogu,
 - `Limit` – kontrola dostępu na podstawie IP (`Deny`, `Allow`, `Order`),
 - `Options` – możliwość używania `Options`,
 - `All` – wszystkie z powyższych,
 - `None` – żadne z powyższych.

Konfiguracja autentykacji

Przykład 1 dla zawartości pliku .htaccess:

```
AuthName "Podaj hasło"
```

```
Access allow all valid-user
```

Udziela dostępu wszystkim użytkownikom z dowolnego adresu IP, po poprawnej autoryzacji za pomocą nazwy użytkownika i hasła, (ustawienia ukrytego katalogu).

Przykład 2 dla zawartości pliku .htaccess:

```
AuthName "Strefa chroniona"
```

```
Access allow all users janek zosia
```

Udziela dostępu z dowolnego adresu IP użytkownikom "janek" oraz "zosia".

Konfiguracja autentykacji

Przykład 3 dla zawartości pliku .htaccess:

```
AuthName "Identyfikacja"
```

```
Access allow 212.85.112.3
```

Udziela dostępu użytkownikowi łączącemu się z konkretnego IP (212.85.112.3), odmawiając innym tego dostępu.

Przykład 4 dla zawartości pliku .htaccess:

```
AuthName "X-Files"
```

```
Access allow all groups wtajemniczeni
```

Udziela dostępu łączącemu się z dowolnego IP i jednocześnie autoryzującego się loginem należącym do grupy "wtajemniczeni". W tym przypadku nazwa użytkownika oraz hasło muszą być dodane do grupy „wtajemniczeni”, aby móc poprawnie przejść proces autoryzacji.

Konfiguracja autentykacji

- Przykład kontrolowania dostępu z użyciem hasła

```
# .htaccess
AuthType Basic # lub Digest
AuthName "Secret"
AuthUserFile password_file
Require user dummy spamer # lub "valid -user „
```

- Plik z hasłami password_file
 - format – użytkownik:hasło
 - hasło szyfrowane MD5 lub crypt
 - generowanie lub zmiana hasła htpasswd2

Konfiguracja PHP

- Instalacja PHP5 i modułu Apache PHP5

```
apt-get install php5 libapache2-mod-php5
```

- mod_perl - Pozwala na wykorzystanie interpretera Perl poprzez uruchomiony proces httpd w celu redukcji tworzenia procesów.
- mod_php - Przyspiesza parsowanie stron zawierających kod PHP.
- mod_speling – naprawia błędy ortograficzne i literówki w URL, które mógł wprowadzić użytkownik

- Restart serwera Apache

```
service apache2 restart
```

- Test PHP5

```
vi /var/www/html/info.php
```

```
<?php
```

```
    phpinfo();
```

```
?>
```

```
http://192.168.0.100/info.php
```

Serwery wirtualne

- Apache udostępnia wiele opcji wirtualnego hostingu
 - /etc/apache2/sites-available
- Zalecane jest używanie raczej IP niż nazwy

```
<VirtualHost 192.168.0.1>
```

```
ServerAdmin webmaster@another-example.org
```

```
DocumentRoot /www/docs/another-example.org
```

```
ServerName www.another-example.org
```

```
ErrorLog logs/another-example.org-error_log
```

```
</VirtualHost>
```

Apache serwery wirtualne – przykłady

```
<VirtualHost firma.com.pl>  
ServerAdmin administartork@firma.com.pl  
DocumentRoot /usr/Apache2/htdocs  
ServerName firma.com.pl  
(...) # tu inne parametry  
ErrorLog /usr/Apache2/logs/error_log  
TransferLog /usr/Apache2/logs/access_log  
</VirtualHost>
```

```
<VirtualHost biuro.firma.com.pl>  
ServerAdmin administartor@firma.com.pl  
DocumentRoot /usr/www/biuro  
ServerName biuro.firma.com.pl  
(...) # tu inne parametry  
ErrorLog /usr/www/biuro/logs/error_log  
TransferLog /usr/www/biuro/logs/access_log  
</VirtualHost>
```

Serwery Wirtualne można tworzyć na podstawie adresów IP czyli do jednego urządzenia sieciowego przypisanych jest kilka adresów IP z odpowiadającym innym adresem domenowym lub poprzez identyfikowanie serwerów różnymi adresami URL związanymi z tym samym adresem IP.

Rozwiązanie wykorzystujące identyfikacje serwerów poprzez nazwy domenowe jest najbardziej polecane i bazuje ono na protokole HTTP/1.1. Rozwiązanie to wykorzystuje jeden adres IP, a rozróżnianie polega na zdefiniowaniu różnych nazw domeny np. *forma.com.pl* oraz *biuro.firma.com.pl* oczywiście nazwy te muszą być zarejestrowane w systemie DNS.

Apache serwery wirtualne – przykłady

```
<VirtualHost 83.210.159.2>  
ServerAdmin administartork@firma.com.pl  
DocumentRoot /usr/Apache2/htdocs  
SerwerName firma.com.pl  
ErrorLog /usr/Apache2/logs/error_log  
TransferLog /usr/Apache2/logs/access_log  
</VirtualHost>
```

```
<VirtualHost 83.210.159.3>  
ServerAdmin administartor@firma.com.pl  
DocumentRoot /usr/www/biuro  
SerwerName biuro.firma.com.pl  
ErrorLog /usr/www/biuro/logs/error_log  
TransferLog /usr/www/biuro/logs/access_log  
</VirtualHost>
```

Identyfikacja serwerów wirtualnych za pomocą adresów IP jest w Internecie ciągle dość popularna mimo iż dostępny zakres adresów drastycznie zmalał. W protokole HTTP/1.0 identyfikacja stron możliwa była jedynie w oparciu o numery IP. Dopiero protokół HTTP/1.1 wprowadził metodę umożliwiającą przesłanie przez przeglądarkę nazwy domenowej serwera. Wielu administratorów korzysta z adresu IP ze względu na to, że starsze przeglądarki nie będą udostępniały możliwości dostępu do serwera Apache

Apache serwery wirtualne – przykłady

Listen 8080

<VirtualHost 83.210.159.2:80>

ServerAdmin administartork@firma.com.pl

DocumentRoot /usr/Apache2/htdocs

SerwerName firma.com.pl

ErrorLog /usr/Apache2/logs/error_log

TransferLog /usr/Apache2/logs/access_log

</VirtualHost>

<VirtualHost 83.210.159.2:8080>

ServerAdmin administartor@firma.com.pl

DocumentRoot /usr/www/biuro

SerwerName biuro.firma.com.pl

ErrorLog /usr/www/biuro/logs/error_log

TransferLog /usr/www/biuro/logs/access_log

</VirtualHost>

Ostatnią metodą konfiguracji serwerów wirtualnych jest identyfikacja oparta na numerach portów. Zaletą tej techniki jest uruchomienie większej ilości serwerów wirtualnych z użyciem jednego adresu IP i pojedynczej nazwy domenowej.

Ze względu na to, że metoda jest oparta na metodzie opierającej się na adresach IP do obsługi nie jest potrzebny protokół HTTP/1.1.

Jeżeli będziemy chcieli obejrzeć stronę biuro.firma.pl, to w przeglądarce musimy podać adres: firma.pl:8080.

Z tego powodu ta metoda nie jest zbyt popularna wśród użytkowników Internetu.

Apache serwery wirtualne – przykłady

```
<VirtualHost 83.210.159.2>  
ServerAdmin administrartork@firma.com.pl  
DocumentRoot /usr/Apache2/htdocs  
ServerName firma.com.pl  
ServerAlias www.firma.pl  
ErrorLog /usr/Apache2/logs/error_log  
TransferLog /usr/Apache2/logs/access_log  
</VirtualHost>
```

Czasami istnieje potrzeba, by dany serwer wirtualny był widoczny pod wieloma nazwami ze względu na przyzwyczajenia użytkowników podczas wpisywania adresów w przeglądarce. Najczęstszym przypadkiem jest wpisywanie adresu bez znaków WWW. Rozwiązanie tego problemu stanowi dyrektywa `ServerAlias`, która umożliwia zdefiniowanie aliasów identyfikujących serwer wirtualny.

Certyfikaty SSL (HTTPS)

- Secure Socket Layer (SSL) port 443
- SSL ochrona komunikacji między przeglądarką a web-serwerem
- Wymaga stworzenia i podpisania certyfikatów (CSR)
- Integralność certyfikatu jest potwierdzana przez urząd Certificate Authority (CA)
- Własny podpis będzie uznawany ale przeglądarka będzie informować o błędzie

Jak działa SSL

- Certyfikaty posiada klucze publiczny i prywatny
- Klucz publiczny – szyfrowanie informacji
- Klucz prywatny – deszyfrowanie
- Klucz prywatny powinien być tajny

Rola Certificate Authority

- Ograniczona liczba instytucji poświadczająca certyfikaty
- Przeglądarki posiadają klucze publiczne
- Walidacja informacji zaszyfrowanej kluczem publicznym
- Przeglądarka deszyfruje i sprawdza czy certyfikat jest podpisany przez CA

KONIEC

