

Tomasz "Zyx" Jędrzejewski

Dynamiczne subdomeny WWW

Wersja 1.0 (3.06.2006)



Szczegółowe informacje o licencji znajdują się pod artykułem.

www.zyxist.com

Dynamiczne subdomeny WWW

Subdomeny są wykorzystywane w Internecie na rozmaite sposoby. Oprócz roli czysto segregacyjnej, gdzie subdomena odpowiada pojedynczemu działowi strony, używa się ich także w charakterze odnośników do znajdujących się na serwerach kont. Użytkownik, rejestrując się w serwisie, dostaje adres np. "login.jakisserwer.pl" oraz konto FTP, przez które może na nie wgrywać dane. Jednak każdy, kto zna podstawy obsługi Bind'a oraz Apache'a, może być nieco skonsternowany, jak taki efekt się osiąga. Obie aplikacje mają konfigurację opartą o pliki tekstowe. Dopisywanie do nich nowych informacji, a potem restartowanie przynajmniej jednego z nich jest przecież nie do pomyślenia. Artykuł ten przybliży bardzo elastyczną metodę zarządzania kontami na własnym serwerze. Zakładam, że czytelnik zna podstawy obsługi programu Bind 9 oraz Apache 2.0, zaś wszystko zostaje pokazane na przykładzie systemu operacyjnego Linux.

Wirtualne hosty

Większość domowych użytkowników serwera Apache wykorzystuje go do zarządzania pojedynczą stronką umieszczoną w jakimś dziwnym miejscu (np. `/var/www` albo nawet katalog `htdocs`). Wszystkie nadesłane żądania obsługiwane są wtedy tak, jakby dotyczyły tego właśnie katalogu. Czasem dochodzi do tego opcja `UserDir` pozwalająca na skonfigurowanie katalogu kont, do których dostęp jest za pomocą tyldy: `www.serwer.pl/~uzytkownik`. Apache posiada jednak znacznie większe możliwości. Jedną kopią tego programu może obsługiwać wiele różnych stron WWW za pomocą tzw. wirtualnych hostów (ang. *virtual hosts*). Termin ten określa metody obsługi kilku różnych witryn przez pojedynczy komputer.

Poniższy przykład pokazuje, jak skonfigurować serwer WWW do obsługi dwóch różnych witryn. Kod ten znajduje się w pliku `httpd.conf` w konfiguracji Apache'a:

```
# Witryna WWW serwera
<VirtualHost www.mojserwer.pl>
    ServerName www.mojserwer.pl
    DocumentRoot /home/users/serwer/www/
    ErrorLog /home/users/serwer/logs/error_log
</VirtualHost>
# Strona domowa
<VirtualHost www.mojastrona.pl>
    ServerName www.mojastrona.pl
    DocumentRoot /home/users/ja/www/
    ErrorLog /home/users/ja/logs/error_log
</VirtualHost>
```

Tak skonfigurowaliśmy dwa wirtualne hosty dla naszych dwóch stron. W każdym z nich podajemy nazwę serwera, katalog z plikami witryny oraz ścieżkę dziennika błędów. Opcji jest znacznie więcej (np. można skonfigurować osobne komunikaty błędów HTTP dla każdej z podstron, czy e-mail administratora). Trzeba tu zwrócić uwagę na pewien istotny mankament powyższego rozwiązania. Każda z naszych podstron wymaga osobnego adresu IP, a to w przypadku większego ruchu jest niemożliwe. Jest tak, ponieważ tak naprawdę żądanie HTTP zawiera sam adres IP serwera i nie niesie ze sobą żadnych informacji o wykorzystanej domenie, która tłumaczona jest na komputerze użytkownika w indywidualnym zakresie przez przeglądarkę. Tak w ogóle przy takich ustawieniach w znaczniku `<VirtualHost>` zaleca się pisanie właśnie adresu IP, ale jako że to jest artykuł, nie chcę tu reklamować czyjś połączenia z siecią :).

Wirtualne hosty bazujące na nazwie

Zaprzecę teraz trochę temu, co pisałem powyżej. Konkretniej mam na myśli fragment mówiący, że w żądaniu HTTP nie ma wzmianki o domenie. Nie jest to do końca prawda. Wszystko zależy po prostu od przeglądarki. Okazuje się, że praktycznie wszystkie będące obecnie w użyciu zachowują nazwę domeny w żądaniu jako informację dodatkową. Apache potrafi to wykorzystać i tu stykamy się z wirtualnymi hostami bazującymi na nazwie domeny (ang. *name-based virtual hosts*). Kłopot z ich obsługą mają jedynie bardzo stare przeglądarki, niewysyłające potrzebnych danych. Używający ich ludzie zawsze zobaczą stronę główną serwera bez względu na wpisywany adres WWW. Aby aktywować ten sposób obsługi wirtualnych hostów,

wystarczy dokleić przed ich deklaracją jedną dyrektywę:

```
NameVirtualHost adres_ip
```

Teraz jesteśmy gotowi do obsługi kilku różnych witryn z różnymi domenami, współdzielącymi jeden adres IP.

Jeżeli pragniemy zachować kompatybilność z antykami, możemy dodać także do konfiguracji każdego wirtualnego hosta linijkę:

```
ServerPath /podstrona
```

Aplikując to do podanych wyżej ustawień, będziemy mogli odwołać się do naszej strony domowej na dwa sposoby: *www.mojastrona.pl* oraz *www.mojserwer.pl/mojastrona*. Zastanów się, czy w ogóle warto się tym zajmować. Autorzy dokumentacji Apache'a twierdzą, że pod słowem "old clients" rozumieją naprawdę stare przeglądarki i wyrażają swą wątpliwość, czy na świecie jest na tyle dużo zdrowych na umyśle ludzi, którzy jeszcze ich używają.

Konfiguracja bez dopisywanek

Z powyższą konfiguracją jest taki kłopot, że nie da się jej zastosować na skalę przemysłową. Zarządzanie megaplikiem *httpd.conf* oraz równie rozbudowanym plikiem strefy w Bindzie zabiłoby większość serwerów. Gościnnie świadczę usługi administracyjne szkolnemu serwerowi i trapiło mnie, jak sprawić, aby każdy uczeń mógł odstać się do swojego konta poprzez intuicyjne *http://login.serwer.pl* zamiast bawić się z tyldami. Drugą przyczyną była wzrastająca ilość subdomen szkolnej witryny WWW. Edytowanie konfiguracji oraz rozrzucenie katalogów po dysku były bardzo męczące. Postanowiłem ubić kilka pieczeni na jednym ogniu i przy okazji ujednocilić zarządzanie podstronami głównej witryny. Poszczególne subdomeny miały stać się zwyczajnymi użytkownikami serwera z własnym kontem FTP oraz katalogiem domowym ulokowanym normalnie w */home*, co ukróciłoby samowolkę i brak jakiegokolwiek systematyczności. Ponadto, dzięki potraktowaniu tego tak samo, jak użytkowników, automatycznie tworzyłaby się obsługa subdomeny. Trapiło mnie tylko, jak to wszystko można wykonać.

Na początku brałem pod uwagę możliwość jakichś tajemnych modyfikacji kodu źródłowego Apache'a oraz Bindy, ale dość szybko to odrzuciłem. Stron korzystających z takiego rozwiązania są tysiące i na pewno ktoś by już dawno zrobił z tego odpowiedni moduł. Dlatego wziąłem na tapetę to, co oba demony nam oferują domyślnie. Nie bawiąc się w opis, jak do czego doszedłem, przechodzę teraz do konkretów.

Od strony Bindy sprawa wygląda prosto. Wystarczy do pliku strefy w miejsce wszystkich subdomen utworzyć jeden rekord CNAME:

```
# CNAME mojserwer.pl.
```

Znak haszu oznacza, że serwer DNS będzie teraz akceptować dosłownie wszystkie nadchodzące subdomeny. Restartujemy demona *named* i już nie musimy bawić się w modyfikowanie konfiguracji Bindy przy dodawaniu nowego użytkownika. Pozostał nam Apache. Tu z pomocą przychodzi nam moduł *mod_vhost_alias*. Na początek upewniamy się, że dyrektywa *UseCanonicalNames* ustawiona jest na "Off". Następnie pod dyrektywą *NamedVirtualHost* dopisujemy

```
VirtualDocumentRoot /home/%-3/www
```

Usuwanie stare wpisy wirtualnych hostów, restartujemy serwer i gotowe. Ciąg *%-3* oznacza odwołanie się do odpowiedniej części wpisanego przez internautę adresu WWW, w tym przypadku loginu zawartego w domenie: *login.mojserwer.pl*. Ciąg *%0* przechowuje całą nazwę domeny. Teraz cała zabawa z wirtualnymi hostami ogranicza się jedynie do utworzenia systemowego użytkownika oraz jego katalogu domowego. Wszystko odbyło się bez napisania ani jednej linijki kodu, czy to w Bashu, czy w C.

Zakończenie

Dzięki dynamicznym wirtualnym hostom wprowadziłem ład w strukturze serwera. Mam nadzieję, że artykuł ten przyda się też innym, gdyż prawdę mówiąc nawet na angielskich stronach ciężko mi było początkowo znaleźć o tym jakiegokolwiek użyteczne informacje (o polskich nawet nie ma co wspominać). Życzę przyjemnego konfigurowania. Bardziej szczegółowych informacji o wirtualnych hostach udzieli Ci dokumentacja Apache'a znajdująca się na stronie <http://httpd.apache.org>.

Tomasz "Zyx" Jędrzejewski

Aktualna wersja artykułu zawsze na **www.zyxist.com**

Licencja

Artykuł rozpowszechniany jest na licencji **Creative Commons Uznanie autorstwa-Użycie niekomercyjne-Bez utworów zależnych 2.5 Polska**.

Wolno:

- kopiować, rozpowszechniać, odtwarzać i wykonywać utwór

Na następujących warunkach:

1. *Uznanie autorstwa*. Utwór należy oznaczyć w sposób określony przez Twórcę lub Licencjodawcę*.
2. *Użycie niekomercyjne*. Nie wolno używać tego utworu do celów komercyjnych.
3. *Bez utworów zależnych*. Nie wolno zmieniać, przekształcać ani tworzyć nowych dzieł na podstawie tego utworu.

W celu ponownego użycia utworu lub rozpowszechniania utworu należy wyjaśnić innym warunki licencji, na której udostępnia się utwór.

Każdy z tych warunków może zostać uchylony, jeśli uzyska się zezwolenie właściciela praw autorskich.

Powyższe postanowienia w żaden sposób nie naruszają uprawnień wynikających z dozwolonego użytku ani żadnych innych praw.

Internetowa skrócona wersja licencji:

<http://creativecommons.org/licenses/by-nc-nd/2.5/pl/>

Pełen tekst licencji:

<http://creativecommons.org/licenses/by-nc-nd/2.5/pl/legalcode>

Podczas publikacji należy podać następujące informacje:

1. Link do internetowej skróconej wersji licencji.
2. Informację o wersji publikowanego tekstu.
3. Informacje o autorze w następujący sposób:

Tomasz "Zyx" Jędrzejewski

Aktualna wersja artykułu zawsze na **www.zyxist.com**