Konfigurowanie serwera WWW Apache na systemie Ubuntu lub Debian VPS

Wprowadzenie

Apache jest obecnie najczęściej używanym serwerem WWW na świecie. Jest to oprogramowanie typu ope source opracowane ponad dwadzieścia lat temu i utrzymywane przez Apache Software Foundation . Szacu się, że od września 2020 r. obsługuje 35% całego ruchu w witrynach internetowych na świecie .

Programiści preferują Apache ze względu na jego szybkość, bezpieczeństwo, niezawodność, solidność i łatwość dostosowywania. Ponadto Apache jest w stanie zaspokoić potrzeby wielu środowisk, ponieważ umożliwia instalację różnych rozszerzeń i modułów. Jest to możliwe dzięki zestawowi konfiguracji, które programiści i administratorzy serwerów mogą modyfikować, aby służyły różnym celom. W tym samouczku przeprowadzimy Cię przez **ogólne pliki konfiguracyjne i opcje, które mogą wpływać na różne wyniki n serwerach Ubuntu i Debian.**

Debian i Ubuntu należą do najczęściej wybieranych systemów operacyjnych dla serwerów. Obaj mają wiele podobieństw. Na przykład oba używają menedżera pakietów APT i pakietów DEB do ręcznej instalacji. Obsługują ustawienia Apache i strukturę katalogów w podobny sposób. Dlatego powinieneś być w stanie wykonać ten samouczek z dowolnym systemem operacyjnym. Najpierw możesz wykonać ten samouczek, aby skonfigurować serwer Ubuntu w kilku krokach .

Następnie zaloguj się jako użytkownik inny niż root z uprawnieniami sudo i zaczynajmy!

Jak zainstalować Apache na Ubuntu i Debianie

Jeśli Twój serwer nie ma jeszcze Apache, możesz go łatwo zainstalować za pomocą jednego polecenia. Najpierw uruchom polecenie aktualizacji:

```
1 sudo apt-get update
```

Następnie możesz zainstalować Apache, wydając następujące polecenie:

```
1 sudo apt-get install apache2
```

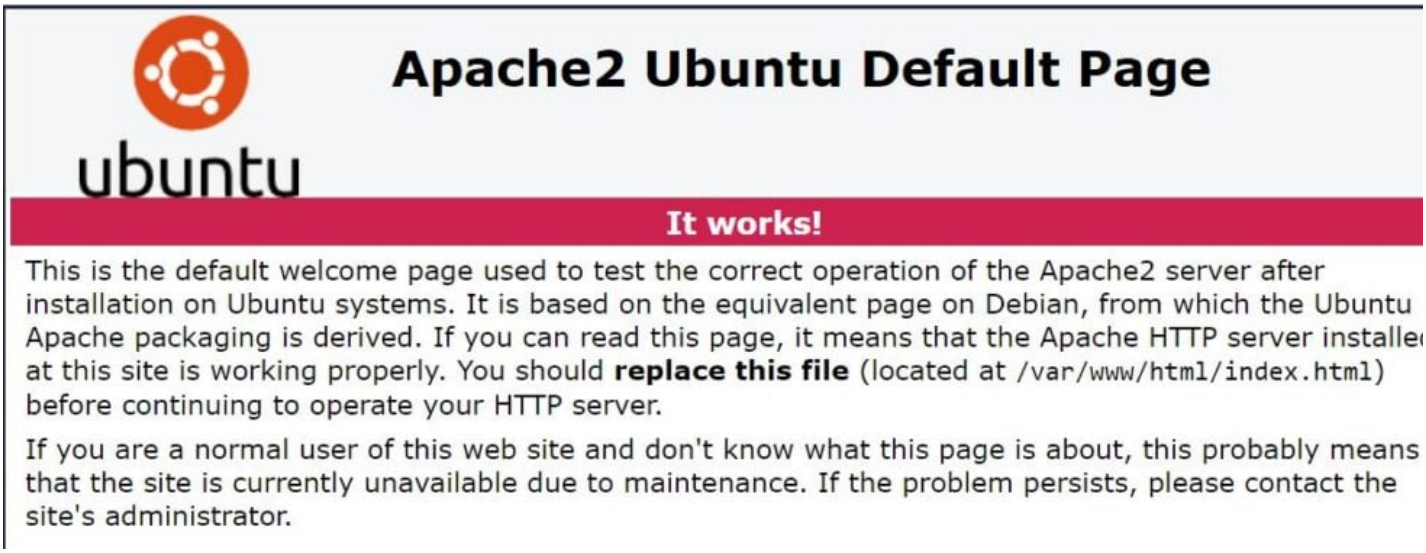
Po zakończeniu instalacji możesz sprawdzić zainstalowaną wersję za pomocą polecenia:

```
1 apache2 -v
```

W momencie pisania tego samouczka wersja była następująca:

```
root@mserver:/home/cloudsigma# apache2 -
Server version: Apache/2.4.41 (Ubuntu)
Server built:   2020-08-12T19:46:17
root@mserver:/home/cloudsigma#
```

Następnie odwiedź adres IP lub nazwę domeny swojego serwera. Powinieneś być w stanie zobaczyć domy stronę indeksu Apache:



Hierarchia plików Apache w dystrybucjach Ubuntu i Debian

Zarówno dystrybucje Ubuntu, jak i Debian używają menedżera pakietów apt . Podobnie, gdy instalujesz pal tworzy on podobne hierarchie plików. W naszym przypadku, zarówno dla Ubuntu, jak i Debiana, Apache przechowuje swoje pliki konfiguracyjne w katalogu „ **/etc/apache2** ”. Przejdź do katalogu za pomocą następującego polecenia i wyświetl zawartość katalogu:

```
1 cd /etc/apache2
2 ls -F
```

Możesz zobaczyć dane wyjściowe polecenia na poniższym zrzucie ekranu:

```
root@mserver:/etc/apache2# ls -F
apache2.conf     conf-enabled/  magic           mods-enabled/  sites-availabl
conf-available/  envvars        mods-available/  ports.conf     sites-enabled/
```

Możesz zobaczyć wiele plików i podkatalogów, z którymi powinieneś się zapoznać podczas konfigurowania konfiguracji serwera Apache. Oto ich opis:

- **apache2.conf** – Jest to główny plik konfiguracyjny serwera Apache. Zawiera również wszystkie domyśl konfiguracje. Chociaż prawie wszystkie konfiguracje można wykonać z poziomu tego pliku, zawsze zale się używanie oddzielnych wyznaczonych plików dla uproszczenia i ochrony tego pliku domyślnego prze uszkodzeniem.
- **conf-available** – Jest to katalog zawierający dodatkowe lokalne pliki konfiguracyjne oraz pliki innych aplikacji, które nie są jeszcze powiązane z żadnymi modułami. Konfiguracje w tym katalogu nie są aktyw chyba że je włączysz. Polecenie **a2enconf** może włączyć plik konfiguracyjny. Oto zawartość tego katalogu:

```
root@mserver:/etc/apache2/conf-available# ls -l
total 20
-rw-r--r-- 1 root root  315 Apr 13  2020 charset.conf
-rw-r--r-- 1 root root 3224 Apr 13  2020 localized-error-pages.conf
-rw-r--r-- 1 root root  189 Apr 13  2020 other-vhosts-access-log.co
-rw-r--r-- 1 root root 2174 Apr 13  2020 security.conf
-rw-r--r-- 1 root root  455 Apr 13  2020 serve-cgi-bin.conf
```

wpływają na zachowanie Apache. Możesz wyłączyć plik konfiguracyjny za pomocą polecenia **a2disconf** . Oto zawartość tego katalogu:

```
root@mserver:/etc/apache2/conf-enabled# ls -l
total 0
lrwxrwxrwx 1 root root 30 Feb 13 12:50 charset.conf -> ../conf-available/charset.conf
lrwxrwxrwx 1 root root 44 Feb 13 12:50 localized-error-pages.conf -> ../conf-available/localized-error-pages.conf
lrwxrwxrwx 1 root root 46 Feb 13 12:50 other-vhosts-access-log.conf -> ../conf-available/other-vhosts-access-log.
lrwxrwxrwx 1 root root 31 Feb 13 12:50 security.conf -> ../conf-available/security.conf
lrwxrwxrwx 1 root root 36 Feb 13 12:50 serve-cgi-bin.conf -> ../conf-available/serve-cgi-bin.conf
```

- **envvars** – Jest to plik zawierający zmienne środowiskowe Apache2. Zmienne w tym pliku mają wpływ t na Apache i nie są związane z twoim systemem Linux. Oto zawartość pliku:

```
# /etc/init.d/apache2, /etc/logrotate.d/apache2, etc.
export APACHE_RUN_USER=www-data
export APACHE_RUN_GROUP=www-data
# temporary state file location. This might be changed to /run in Wheez
export APACHE_PID_FILE=/var/run/apache2$SUFFIX/apache2.pid
export APACHE_RUN_DIR=/var/run/apache2$SUFFIX
export APACHE_LOCK_DIR=/var/lock/apache2$SUFFIX
# Only /var/log/apache2 is handled by /etc/logrotate.d/apache2.
export APACHE_LOG_DIR=/var/log/apache2$SUFFIX

## The locale used by some modules like mod_dav
export LANG=C
## Uncomment the following line to use the system default locale instea
#. /etc/default/locale

export LANG

## The command to get the status for 'apache2ctl status'.
## Some packages providing 'www-browser' need '--dump' instead of '-dum
#export APACHE_LYNX='www-browser -dump'
```

- **magic** – Jest to plik zawierający instrukcje określające typ MIME pliku. Pomaga serwerowi określić typ pliku.
- **mods-available** – Jest to katalog zawierający pliki konfiguracyjne do ładowania modułów i ich konfigurowania. Jednak nie są one jeszcze włączone. Możesz je włączyć za pomocą **polecenia a2enm**
- **mods-enabled** – Jest to katalog zawierający pliki konfiguracyjne do ładowania modułów i ich konfigurowania. Konfiguracje w tym katalogu są już włączone i prowadzą do odpowiednich modułów. Możesz wyłączyć moduły za pomocą **polecenia a2dismod.**
- **ports-conf** – Jest to plik określający porty dostępne dla hostów wirtualnych i porty TCP, których nasłuch Apache.
- **sites-available** – Jest to katalog zawierający dostępne pliki konfiguracyjne dla wirtualnych hostów Apache. Wirtualne hosty pozwalają Apache obsługiwać różne strony internetowe. Pliki w tym katalogu n są jeszcze aktywne. Możesz włączyć wirtualny plik hosta za pomocą polecenia **a2ensite** .
- **sites-enabled** – Jest to katalog zawierający aktywowane pliki konfiguracyjne wirtualnego hosta. Zwykle zawiera dowiązania symboliczne do plików w katalogu witryn dostępnych. Możesz wyłączyć wirtualny p hosta za pomocą polecenia **a2dissite** .

Modułowa konstrukcja plików konfiguracyjnych Apache umożliwia modyfikowanie i dodawanie plików według potrzeb.

Zawartość pliku Apache2.conf

Plik w „ **/etc/apache2/apache2.conf** ” zawiera główne pliki konfiguracyjne dla Apache. Mówiąc dokładniej, zawiera dyrektywy konfiguracyjne, które przekazują serwerowi instrukcje. Składa się z trzech sekcji służący do definiowania **globalnych** konfiguracji procesów serwera, **domyślnej** konfiguracji serwera i konfiguracji **hostów wirtualnych** .

W systemach Ubuntu i Debian większość pliku zawiera konfiguracje globalnych procesów serwera. W dolne części można znaleźć konfiguracje domyślnego serwera i hostów wirtualnych za pomocą dyrektywy „ **Inclu ”.** Apache używa dyrektywy „Include” do odczytywania innych plików konfiguracyjnych, a następnie włączan ich do bieżącego pliku konfiguracyjnego od miejsca, w którym pojawiła się dyrektywa. Dzieje się tak podcza uruchamiania serwera Apache.

Przewijając w dół pliku apache2.conf, powinieneś zobaczyć kilka dyrektyw dołączania. Dyrektywy ładują definicje modułów w katalogu **mods-enabled** , plik **ports.conf** , lokalne konfiguracje w katalogu **conf-enabled** oraz konfiguracje hostów wirtualnych w katalogu site **-enabled** .

Sekcja globalnej konfiguracji serwera

Sekcja globalnej konfiguracji serwera zawiera ustawienia i instrukcje kontrolujące globalne działanie Apache. Zawiera kilka dyrektyw, których wartości można modyfikować, aby zmienić zachowanie Apache. Omówmy niektóre z nich:

- **Timeout** – Ta dyrektywa określa maksymalny czas, w którym serwer może spełnić żądanie. Domyślnie to 300 sekund, co jest prawdopodobnie zbyt wysokie, jeśli tworzysz szybciej responsywną aplikację. Możesz zmienić tę wartość na niższą, jeśli serwer ma wystarczającą ilość zasobów.
- **KeepAlive** — określa, że każde połączenie powinno pozostać otwarte w celu obsługi wielu żądań tego samego klienta. Domyślnie jest ustawiony na „Wł.”. Jeśli jednak jest ustawiony na „Wył.”, każde żądanie będzie musiało ustanowić nowe połączenie, nawet jeśli pochodzi od tego samego klienta. Może to spowodować znaczne przeciążenie serwera.
- **MaxKeepAliveRequests** — Kontroluje maksymalną liczbę żądań, które można zezwolić podczas jedne trwałego połączenia przed śmiercią. Aby zezwolić na nieograniczoną liczbę połączeń, ustaw wartość na Aby uzyskać maksymalną wydajność i wydajność, ustaw najwyższą możliwą liczbę.
- **KeepAliveTimeout** — określa maksymalną liczbę sekund oczekiwania na następne żądanie od tego samego klienta po ostatnim przed zerwaniem połączenia. Wartość domyślna to 5 sekund. Jeśli czas upłynie, to w następnym żądaniu nawiązywane jest nowe połączenie w celu obsługi żądania.
- **HostnameLookups** – Decyduje, czy wyszukiwanie DNS powinno być włączone, czy nie. Domyślnie jes wyłączone. Jeśli ją włączysz, każde żądanie spowoduje co najmniej jedno wyszukanie serwera nazw, c znacznie negatywnie wpłynie na wydajność serwera.

Domyślny plik hosta wirtualnego

Świeża instalacja serwera Apache zawiera domyślny wirtualny plik hosta o nazwie **000-default.conf** . Jest t plik obsługujący domyślną stronę Apache index.html. Znajduje się w katalogu dostępnych witryn. Zawartość tego pliku daje ogólne wyobrażenie o formacie plików wirtualnego hosta Apache. Wprowadź następujące polecenie, aby otworzyć plik:

```
sudo nano /etc/apache2/sites-available/000-default.conf
```

Oto zawartość pliku. Usunęliśmy komentarze dla zwięzłości:

```
<VirtualHost *:80>

        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html

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

```

Jak widać w nagłówku konfiguracji **VirtualHost *:80** , domyślnie ten wirtualny host jest skonfigurowany do obsługi żądań na porcie 80. Jest to standardowy port HTTP.

Apache został zaprojektowany tak, aby używał najbardziej szczegółowej definicji hosta wirtualnego, która pasuje do konkretnego żądania. Oznacza to, że ten konkretny wirtualny host niekoniecznie musi obsłużyć każde żądanie do serwera. Zamiast tego, gdyby istniał inny plik wirtualnego hosta z bardziej szczegółową definicją, zastąpiłby ten plik.

Oto opisy dyrektyw w pliku hosta wirtualnego:

- **ServerAdmin** — Określa kontaktowy adres e-mail używany w przypadku problemów z serwerem. Apac zwykle wyświetla to na wygenerowanej stronie błędu. Jeśli ustawisz **ServerSignature** na „Email” w **plik /etc/apache2/conf-enabled/security.conf** , podany tutaj adres e-mail **będzie**
- **DocumentRoot** – Ta dyrektywa wskazuje katalog zawierający pliki strony internetowej. Domyślną stron indeksu powitalnego Apache można znaleźć w katalogu **/var/www/html** w Ubuntu 20.04.
- **ErrorLog** — ta dyrektywa określa lokalizację pliku dziennika błędów.
- **CustomLog** — ta dyrektywa pokazuje lokalizację dziennika dostępu i format wyświetlania dziennika.

Dodatkowe dyrektywy hosta wirtualnego

- **ServerName** – Określa nazwę domeny lub adres IP, który powinien obsługiwać wirtualny host. Gdybyśr definiowali wiele hostów wirtualnych, aby nasz serwer Apache mógł obsługiwać wiele witryn internetowy dodalibyśmy dyrektywę ServerName i określilibyśmy domenę, którą powinien obsługiwać każdy wirtualr host.
- **ServerAlias** — ta dyrektywa powoduje, że jeden plik wirtualnego hosta ma zastosowanie do więcej niż jednej nazwy, zapewniając alternatywne domeny umożliwiające dostęp do tej samej treści. Na przykład, Twoi użytkownicy mają dostęp do tych samych treści, jeśli poprzedzą nazwę Twojej domeny **przedrostkiem www** . Na przykład możemy dodać następujący element do hosta wirtualnego dla domeny **example.com :**

```
ServerName example.com
ServerAlias www.example.com
```

Dyrektywa dotycząca definicji katalogów

W rzeczywistych przypadkach prawdopodobnie będziesz mieć pliki swojej witryny w kilku katalogach. Domyślny plik hosta wirtualnego nie ma dyrektywy **Directory** , ponieważ może obsługiwać tylk jeden plik index.html. Dyrektywa **Directory** określa, w jaki sposób Apache obsługuje różne katalogi w katalo głównym dokumentu witryny. W tej dyrektywie definiujesz również ograniczenia dostępu do systemu plików. Domyślnie Apache umożliwia dostęp do Internetu tylko do plików w katalogu **/var/www** . W tym kata możesz utworzyć inne katalogi, w których będą przechowywane pliki Twojej witryny. Ponadto możesz określ w jaki sposób chcesz uzyskać dostęp do plików w katalogu. Przykładowa definicja może wyglądać tak:

```
<Directory /var/www/html>
    Options Indexes FollowSymLinks MultiViews
    Require all granted
    AllowOverride all
</Directory>
```

< Katalog **/var/www/html>** określa katalog, do którego będą miały zastosowanie te reguły. Opcja „ **AllowOverride** ” określa, czy plik „ **.htaccess** ” powinien zastąpić ustawienia, jeśli umieścisz go w określo katalogu. Jest to konieczne podczas pracy z frameworkami takimi jak Laravel, które współpracują z .htacce przepisywaniem adresów URL.

Aby dowiedzieć się więcej o wirtualnych hostach, możesz odwiedzić nasz samouczek dotyczący konfigurowania wirtualnych hostów Apache w systemie Ubuntu 20.04 . Oto przykładowy plik wirtualnego hosta ze wszystkimi zdefiniowanymi przez nas dyrektywami:

```
<VirtualHost *:80>
    ServerName example.com
    ServerAlias www.example.com
    ServerAdmin webmaster@localhost.com
    DocumentRoot /var/www/html
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined


        <Directory /var/www/html>
                Options Indexes FollowSymLinks MultiViews
                Require all granted
                AllowOverride all
        </Directory>
</VirtualHost>
```

Dyrektywy przekierowania, Alias i ScriptAlias

W niektórych przypadkach możesz chcieć uzyskać dostęp do plików witryny lub plików wykonywalnych, któ nie znajdują się bezpośrednio w katalogu DocumentRoot. I tu wkraczają te trzy dyrektywy .

Dyrektywa **Redirect** nakazuje klientom wysłanie nowego żądania z innym adresem URL. Zwykle użyjesz g gdy będziesz chciał przekierować odwiedzających Twoją stronę internetową na inną nazwę domeny lub ad IP. Zauważ, że **Redirect** jest najpierw wykonywany przed Alias i ScriptAlias. Dyrektywa **Alias** odwzorowuje adres URL na ścieżkę do katalogu. Dyrektywa **ScriptAlias** odwzorowuje adres URL na katalogi, które będą zawierać skrypty wykonywalne. Aby Alias i ScriptAlias działały, musisz ustawić uprawnienia do katalogu dla użytkownika danych www, z którego korzysta sieć.

Jak włączyć witryny i moduły w Apache

Wirtualne hosty i moduły muszą zostać włączone, zanim staną się aktywne i dostępne dla procesu Apache. Ponadto wirtualne hosty są tworzone w katalogu **witryn dostępnych** , jak wyjaśniono. Jeśli masz wirtualnego hosta o nazwie **example.com.conf** , możesz go włączyć za pomocą następującego polecenia:

```
sudo a2ensite example.com.conf
```

Polecenie włącza wirtualnego hosta. Jednak aby zmiany odniosły skutek, musisz poinstruować Apache, ab przeładował pliki konfiguracyjne, wprowadzając następującą komendę:

```
sudo service apache2 reload
```

Możesz chcieć wyłączyć wirtualny plik hosta o nazwie **example.com.conf** , aby Apache nie używał go do obsługi żądań. Aby to zrobić, wprowadź następujące polecenie, aby wyłączyć plik:

```
sudo a2dissite example.com.conf
```

Ponownie, aby zmiany zaczęły obowiązywać natychmiast, musisz poinstruować Apache, aby przeładował p konfiguracyjne, wprowadzając następującą komendę:

```
sudo service apache2 reload
```

Podobnie możesz włączać i wyłączać moduły przy użyciu tej samej składni. Polecenie włączania modułu to „ **a2enmod** ”, a polecenie wyłączania modułu to „ **a2dismod”.** ” Na przykład istnieje moduł Apache o nazwie mod_rewrite , który pomaga w przepisywaniu adresów URL w zależności od konfiguracji .htaccess. Musisz włączyć ten moduł, aby plik .htaccess działał. Możesz to zrobić wpisując polecenie:

```
sudo a2enmod rewrite
```

Aby ta zmiana zaczęła obowiązywać, musisz ponownie załadować konfigurację Apache. Możesz również

```
sudo a2dismod rewrite
```

Następnie ponownie załaduj konfiguracje Apache2, aby zmiany zaczęły obowiązywać natychmiast.

Wreszcie, oto kilka dodatkowych zasobów, które pomogą Ci zapoznać się z serwerem WWW Apache:

- Instalowanie serwera Apache na Ubuntu 18.04: poradnik
- Jak zainstalować stos Linux, Apache, MySQL, PHP (LAMP) na CentOS 7
- Zabezpieczanie Apache za pomocą Let's Encrypt na Ubuntu 18.04
- Świat serwerów WWW: Apache kontra Nginx

Wniosek

W tym samouczku zdefiniowaliśmy podstawowe pliki konfiguracyjne Apache w systemach Ubuntu i Debian. wykazaliśmy, Apache jest wysoce modułowy, co oznacza, że można zainstalować wiele modułów do wykonywania różnych zadań.

Możesz łatwo włączać i wyłączać te moduły oraz zmieniać inne konfiguracje, aby osiągnąć różne wyniki. Możesz teraz modyfikować lokalne pliki konfiguracyjne, które są następnie dołączane do głównego konfiguracyjnego. Dzięki temu nie musisz się martwić o uszkodzenie globalnego pliku konfiguracyjnego.