

Konfiguracja wielu stron WWW „opartych na nazwie”

Instalacja Apache2

Aby zainstalować serwer WWW Apache2 dostępny w Ubuntu, w terminalu należy wpisać następujące polecenie:

```
sudo apt-get -y install apache2
```

Podstawy konfiguracji Apache2

Apache2 jest konfigurowany poprzez umieszczanie *dyrektyw* w plikach konfiguracyjnych w postaci zwykłego tekstu. *Dyrektywy* te są rozdzielone między następujące pliki i katalogi z lokalizowane w `/etc/apache2/`:

1. *apache2.conf*: główny plik konfiguracyjny Apache2. Zawiera ustawienia *globalne* dla Apache2.
2. *sites-available*: ten katalog zawiera pliki konfiguracyjne dla *wirtualnych hostów* Apache2. Wirtualne hosty pozwalają na konfigurację Apache2 dla wielu witryn, które mają osobne konfiguracje.
3. *sites-enabled*: przechowuje dowiązania symboliczne do plików w katalogu `/etc/apache2/sites-available`. Gdy plik konfiguracyjny w *sites-available* jest dowiązany symbolicznie, witryna będzie aktywna po ponownym uruchomieniu Apache2.
4. *ports.conf*: zawiera dyrektywy określające, na których portach TCP nasłuchuje Apache2.
5. *mods-available*: ten katalog zawiera pliki konfiguracyjne do załadowania i skonfigurowania *modułów*. Jednak nie wszystkie moduły będą miały określone pliki konfiguracyjne.
6. *mods-enabled*: przechowuje *dowiązania symboliczne* do plików w katalogu `/etc/apache2/mods-available`. Kiedy plik konfiguracyjny modułu zostanie dowiązany symbolicznie, zostanie włączony przy następnym uruchomieniu *apache2*.
7. *conf-available*: ten katalog zawiera dostępne pliki konfiguracyjne.
8. *conf-enabled*: przechowuje *dowiązania symboliczne* do plików w katalogu `/etc/apache2/conf-available`. Kiedy plik konfiguracyjny zostanie dowiązany symbolicznie, zostanie włączony przy następnym uruchomieniu *apache2*.
9. *envvars*: plik, w którym ustawione są zmienne *środowiskowe* Apache2.
10. *magic*: instrukcje określania typu MIME na podstawie pierwszych kilku bajtów pliku.

Ponadto inne pliki konfiguracyjne można dodawać za pomocą dyrektywy *Include*, a symboli wieloznacznych można używać do dołączania wielu plików konfiguracyjnych. Każda dyrektywa może być umieszczona w dowolnym z tych plików konfiguracyjnych. Zmiany w głównych plikach konfiguracyjnych są rozpoznawane przez Apache2 dopiero po uruchomieniu lub ponownym uruchomieniu.

Serwer odczytuje również plik zawierający typy dokumentów MIME; nazwa pliku jest ustawiana przez dyrektywę *TypesConfig*, zwykle przez `/etc/apache2/mods-available/mime.conf`, która może również zawierać dodatki i przesłonięcia, i domyślnie jest to `/etc/mime.types`.

Konfigurowanie nowej witryny WWW.

Konfigurowanie nowej witryny WWW, polega na utworzeniu i skonfigurowaniu nowego wirtualnego hosta w katalogu *sites-available*. Aby strona była widoczna należy utworzyć dowiązanie symboliczne do pliku konfiguracyjnego nowego wirtualnego hosta, w katalogu *sites-enabled*. Można do tego wykorzystać skrypt **a2ensite**:

```
sudo a2ensite <Nazwa pliku konfiguracyjnego Wirtualnego Hosta>
```

oraz uruchomić ponownie Apache2

```
sudo systemctl restart apache2.service
```

lub

```
sudo service apache2 reload
```

wynik restartu można zweryfikować poleceniami:

```
sudo systemctl status apache2.service
```

lub

```
sudo service apache2 status
```

Aby zatrzymać działanie witryny (strona będzie nie widoczna) należy usunąć dowiązanie symboliczne do pliku konfiguracyjnego wirtualnego hosta, w katalogu sites-enabled. Można do tego wykorzystać skrypt **a2dissite**:

```
sudo a2dissite <Nazwa pliku konfiguracyjnego Wirtualnego Hosta>
```

oraz uruchomić ponownie Apache2

```
sudo systemctl restart apache2.service
```

lub

```
sudo service apache2 restart
```

Podstawowe parametry konfiguracyjne serwera Apache2.

1. Apache2 jest dostarczany z domyślną konfiguracją przyjazną dla wirtualnego hosta. Oznacza to, że jest skonfigurowany z pojedynczym domyślnym hostem wirtualnym (za pomocą dyrektywy *VirtualHost*), który można modyfikować lub używać w stanie, w jakim się znajduje, jeśli masz jedną witrynę lub jako szablon dla dodatkowych hostów wirtualnych, jeśli masz wiele witryn.
2. Dyrektywa *ServerAdmin*: określa adres e-mail administratora serwera. Jeżeli wystąpi problem z witryną, Apache2 wyświetli komunikat o błędzie zawierający ten adres e-mail. Dyrektywa znajduje się w pliku konfiguracyjnym witryny w `/etc/apache2/sites-available`. Wartość domyślna to `webmaster@localhost`.
3. Dyrektywa *Listen* określa port, oraz ewentualnie adres IP, na którym Apache2 powinien nasłuchiwać. Jeśli adres IP nie zostanie określony, Apache2 będzie nasłuchiwał na wszystkich adresach IP przypisanych do komputera, na którym działa. Wartością domyślną dyrektywy *Listen* jest 80. Tę dyrektywę można znaleźć i zmienić w pliku `/etc/apache2/ports.conf`
4. Dyrektywa *ServerName* jest opcjonalna. Określa FQDN witryny. Domyślny host wirtualny nie ma określonej dyrektywy *ServerName*, więc będzie odpowiadał na wszystkie żądania, które nie są zgodne z dyrektywą *ServerName* na innym hoście wirtualnym.
5. Dyrektywa *ServerAlias* jest opcjonalna. Określa aliasy FQDN witryny. Może zawierać symbole wieloznaczne.
6. Dyrektywa *DocumentRoot*: określa gdzie Apache2 powinien szukać plików, które tworzą witrynę. Wartość domyślna to `/var/www/html`, jak określono w `/etc/apache2/sites-available/000-default.conf`. W razie potrzeby można zmienić tę wartość w pliku wirtualnego hosta witryny i należy pamiętać, aby utworzyć ten katalog.

Ustawienia domyślne

Ustawienia skonfigurowane dla hosta wirtualnego mają pierwszeństwo dla tego hosta wirtualnego, przed ustawieniami globalnego. W przypadku dyrektywy niezdefiniowanej w ustawieniach hosta wirtualnego używana jest wartość domyślna.

Prowadzenia więcej niż jednej strony internetowej na jednym komputerze jest ściśle związana z terminem „*Virtual Host*”. Wirtualne hosty mogą być:

- „oparte na adresie IP”, co oznacza, że masz inny adres IP dla każdej strony internetowej, lub
- „oparte na nazwie”, co oznacza, że masz wiele nazw działających na każdym adresie IP.

Fakt, że działają na tym samym serwerze fizycznym, nie jest oczywisty dla użytkownika końcowego.

Konfiguracja wielu stron WWW „opartych na nazwie”

Hosty wirtualne oparte na protokole IP używają adresu IP połączenia, do określenia właściwego hosta wirtualnego do obsługi. Dlatego musisz mieć, dla każdego hosta odrębny adres IP.

W przypadku wirtualnego hostingu **opartego na nazwach** działanie serwera polega na tym, że **klient zgłasza nazwę hosta, jako część nagłówka HTTP**. Korzystając z tej techniki, **wiele różnych hostów może współdzielić ten sam adres IP**.

Wirtualny hosting oparty na nazwach wymaga:

- skonfigurowania serwera DNS, aby mapował każdą nazwę hosta na poprawny adres IP.
- skonfigurowania serwera HTTP Apache, aby rozpoznawał różne nazwy hostów.

Wirtualny hosting oparty na nazwie zmniejsza zapotrzebowanie na rzadkie adresy IP. Dlatego należy używać hostingu wirtualnego opartego na nazwie. Chyba, że używasz sprzętu, który wyraźnie wymaga hostingu opartego na protokole IP. Historyczne powody hostingu wirtualnego opartego na protokole IP nie mają już zastosowania do serwerów WWW ogólnego przeznaczenia.

Hosting wirtualny oparty na nazwie opiera się na algorytmie wyboru hosta wirtualnego opartego na IP. Co oznacza, że wyszukiwanie prawidłowej nazwy serwera odbywa się tylko między hostami wirtualnymi, które mają najlepszy adres oparty na IP.

Jak serwer wybiera odpowiedni host wirtualny oparty na nazwie.

Należy pamiętać, że pierwszym krokiem w rozwiązywaniu hosta wirtualnego opartego na nazwie jest rozpoznawanie oparte na protokole IP. Rozpoznawanie hosta wirtualnego na podstawie nazwy polega na wyborze najbardziej odpowiedniego hosta wirtualnego na podstawie nazwy, po zawężeniu kandydatów do najlepszego dopasowania opartego na adresie IP. Użycie symbolu wieloznacznego (*) dla adresu IP we wszystkich dyrektywach VirtualHost powoduje, że to mapowanie oparte na adresie IP jest nieistotne.

Kiedy nadejdzie żądanie, serwer znajdzie najlepszy (najbardziej konkretny) **<VirtualHost>** na podstawie adresu IP i portu używanego przez żądanie. Jeśli istnieje więcej niż jeden wirtualny host zawierający ten najlepiej dopasowany adres i kombinację portów, Apache dokona dalszego porównania dyrektyw **ServerName** i **ServerAlias** z nazwą serwera obecną w żądaniu.

Jeśli pominiesz dyrektywę **ServerName** na dowolnym wirtualnym hoście opartym na nazwie, serwer domyślnie przyjmie w pełni kwalifikowaną nazwę domeny (FQDN) pochodzącą z systemowej nazwy hosta. Ta niejawnie ustawiona nazwa serwera może prowadzić do sprzecznego z intuicją dopasowywania hosta wirtualnego i jest odradzana.

Domyślny wirtualny host oparty na nazwie dla kombinacji adresu IP i portu

Jeśli w zestawie wirtualnych hostów zawierających najlepiej dopasowany konkretny adres IP i kombinację portów, nie zostanie znaleziona żadna, zgodna nazwa serwera, wówczas **zostanie wyświetlony pierwszy wymieniony host wirtualny**.

Korzystanie z wirtualnych hostów opartych na nazwach

Pierwszym krokiem jest utworzenie bloku **<VirtualHost>** dla każdego hosta, który chcesz obsługiwać. Wewnątrz każdego bloku **<VirtualHost>** potrzebujesz co najmniej dyrektywy **ServerName**, aby określić, który host jest obsługiwany, oraz dyrektywy **DocumentRoot**, która pokaże, gdzie w systemie plików znajduje się zawartość tego hosta.

Główny host

*Każde żądanie, które nie pasuje do istniejących **<VirtualHost>** jest obsługiwane przez globalną konfigurację serwera, niezależnie od nazwy hosta lub nazwy serwera.*

*Po dodaniu hosta wirtualnego opartego na nazwie do istniejącego serwera, o argumentach zgodnych z istniejącymi kombinacjami adresów IP i portów, żądania będą obsługiwane przez jawny host wirtualny. W takim przypadku zwykle rozsądnie jest utworzyć domyślnego wirtualnego hosta z **ServerName**, pasującym do serwera podstawowego. Nowe domeny na tym samym interfejsie i porcie, wymagające osobnych konfiguracji, można dodać jako kolejne (inne niż domyślne) hosty wirtualne.*

Dziedziczenie ServerName

*Najlepiej jest zawsze jawnie wymieniać nazwę **ServerName**, na każdym hoście wirtualnym opartym na nazwie.*

*Jeśli **VirtualHost** nie określi **ServerName**, nazwa serwera zostanie odziedziczona z konfiguracji serwera podstawowego. Jeśli globalnie nie podano nazwy serwera, zostanie on wykryty podczas uruchamiania poprzez odwrotne rozpoznawanie DNS, pierwszego adresu nasłuchującego. W obu przypadkach, ta odziedziczona nazwa serwera wpłynie na rozpoznawanie hosta wirtualnego na podstawie nazwy, dlatego najlepiej jest zawsze jawnie wymieniać nazwę **ServerName** na każdym hoście wirtualnym opartym na nazwie.*

Założmy na przykład, że obsługujesz domenę `www.example.com` i chcesz dodać hosta wirtualnego `other.example.com`, który wskazuje ten sam adres IP. Dodajesz następujące hosty wirtualne

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

```
# Ten wirtualny host, który znajduje się na liście, jest również domyślny dla *: 80
```

```
    ServerName example.com
```

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

```
    DocumentRoot "/www/domain"
```

```
</VirtualHost>
```

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

```
    ServerName other.example.com
```

```
    DocumentRoot "/www/otherdomain"
```

```
</VirtualHost>
```

Alternatywnie możesz podać jawny adres IP zamiast `*` w dyrektywach **<VirtualHost>**. Na przykład możesz to zrobić, aby uruchomić hosty wirtualne oparte na nazwie na jednym adresie IP, oraz inny zestaw hostów wirtualnych opartych na nazwie na innym adresie IP.

Wiele serwerów może być dostępnych pod więcej niż jedną nazwą. Jest to możliwe dzięki dyrektywie **ServerAlias** umieszczonej wewnątrz sekcji **<VirtualHost>**. Na przykład w pierwszym bloku **<VirtualHost>** powyżej, dyrektywa **ServerAlias** wskazuje, że wymienione nazwy są innymi nazwami, których można użyć, aby zobaczyć tę samą stronę internetową:

Np. **ServerAlias example.com *.example.com**

Wówczas, żądania dotyczące hostów w domenie example.com będą obsługiwane przez wirtualnego hosta www.example.com. Symbole wieloznaczne * i ? mogą być używane do dopasowywania nazw.

Oczywiście nie można po prostu wymyślić nazw i umieścić ich w ServerName i/lub ServerAlias.

Najpierw należy odpowiednio skonfigurować serwer DNS, aby mapował te nazwy na adres IP powiązany z serwerem.

Hosty wirtualne oparte na nazwie dla najlepiej dopasowanego zestawu **<VirtualHost>** są przetwarzane w kolejności, w jakiej występują w konfiguracji. Pierwsze dopasowanie **ServerName** lub **ServerAlias** jest używane, bez pierwszeństwa dla symboli wieloznacznych (ani dla **ServerName** vs. **ServerAlias**).

Pełna lista nazw w dyrektywie **VirtualHost** jest traktowana jak (bez symboli wieloznacznych) **ServerAlias**.

Na koniec możesz dostosować konfigurację wirtualnych hostów, umieszczając inne dyrektywy w kontenerach **<VirtualHost>**. Większość dyrektyw można umieścić w tych kontenerach, a zmieniają one konfigurację tylko odpowiedniego wirtualnego hosta. Aby dowiedzieć się, czy określona dyrektywa jest dozwolona, należy sprawdzić jej kontekst. Dyrektywy konfiguracyjne ustawione w *kontekście głównego serwera* (poza dowolnym kontenerem **<VirtualHost>**) będą używane tylko wtedy, gdy nie zostaną zastąpione przez ustawienia wirtualnego hosta.

Przykład praktyczny.

Zadanie.

Skonfiguruj dwie strony www: zima i wiosna powiązane z adresem 172.22.y.5/24

Witryny utwórz:

- zima: w podkatalogu zima katalogu domowego użytkownika uxyy, czyli /home/uxyy/zima/. Plik startowy zima.html
- wiosna: w podkatalogu wiosna katalogu głównego, czyli /wiosna/. Plik startowy wiosna.html

Witryny powinny otwierać się po nazwach domenowych, odpowiednio:

- zima.info, alias www.zima.info
- wiosna.info, alias www.wiosna.info

Każda strona www powinna zawierać informacje o:

- Adresie Ip powiązanym z witryną
- Nazwie, domenie i aliasach otwierających witrynę
- Metodzie utworzenia strony „oparta na nazwie”
- Lokalizacji plików z zawartością strony
- Jak nazywa się plik startowy.

Założenia

Do zrealizowania zadania i wszystkich przykładów, niezbędny jest:

1. Serwer Ubuntu 20.
 - a. Zainstalowany i skonfigurowany serwer DNS:
 - i. Domena: szkoła.lokalna
 - ii. Adres IP domeny i serwera nazw: 172.22.y.1
 - b. Zainstalowany serwer WWW – Apache2
2. Klient Ubuntu Desktop 20

Przykładowa dokumentacja rozwiązania zadania.

Sprawdzamy, czy na serwerze jest zainstalowany Apache2.

```
dpkg-query -L apache2
```

```
us3n00@ks23-3n00: ~  
us3n00@ks23-3n00:~$ dpkg-query -L apache2  
./.  
/etc  
/etc/apache2  
/etc/apache2/apache2.conf  
/etc/apache2/conf-available  
/etc/apache2/conf-available/charset.conf
```

Sprawdzamy, czy jest skonfigurowany adres 172.22.0.5/24 na karcie sieci lokalnej LAN

```
us3n00@ks23-3n00: /  
us3n00@ks23-3n00:/$ ip a |grep LAN  
3: LAN: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group  
default qlen 1000  
    inet 172.22.0.1/24 brd 172.22.0.255 scope global LAN  
    inet 172.22.0.2/24 brd 172.22.0.255 scope global secondary LAN  
    inet 172.22.0.11/24 brd 172.22.0.255 scope global secondary LAN  
us3n00@ks23-3n00:/$
```

Jak widać nie ma takiego adresu. Należy go dopisać w pliku /etc/netplan/00-installer-config.yaml.

```
Otwórz 00-installer-config.yaml [Tylko do odczytu] Zapisz  
/etc/netplan  
1 # This is the network config written by 'subiquity'  
2 network:  
3   ethernet:  
4     enp0s3:  
5       dhcp4: true  
6       match:  
7         macaddress: 08:00:27:c7:c7:4c  
8         set-name: WAN  
9  
10    enp0s8:  
11      addresses:  
12        - 172.22.0.1/24  
13        - 172.22.0.2/24  
14        - 172.22.0.5/24  
15        - 172.22.0.11/24  
16      match:  
17        macaddress: 08:00:27:75:31:17  
18        set-name: LAN  
19    version: 2  
20
```

Po restarcie usługi

```
sudo netplan apply
```

Sprawdzamy konfigurację interfejsu LAN

```
us3n00@ks23-3n00: /  
us3n00@ks23-3n00:/$ sudo netplan apply  
us3n00@ks23-3n00:/$ ip a |grep LAN  
3: LAN: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default  
qlen 1000  
    inet 172.22.0.1/24 brd 172.22.0.255 scope global LAN  
    inet 172.22.0.2/24 brd 172.22.0.255 scope global secondary LAN  
    inet 172.22.0.11/24 brd 172.22.0.255 scope global secondary LAN  
    inet 172.22.0.5/24 brd 172.22.0.255 scope global secondary LAN  
us3n00@ks23-3n00:/$
```

Na serwerze głównym z domeną podstawową **szkola.lokalna**, tworzymy dwie strefy wyszukiwania do przodu:

1. **zima.info**
2. **wiosna.info**

dopisując do pliku **/etc/bind/named.conf.local**, wiersze definiujące nowe strefy wyszukiwania do przodu.

Tworzymy nowe strefy klauzulą **zone**, definiujemy, że są to strefy główne {master}. Określamy również, lokalizację plików stref - katalog **/etc/bind/**, w których utworzymy wymagane rekordy DNS.

```

us3n00@ks23-3n00: /
us3n00@ks23-3n00:/$ tail -n 11 /etc/bind/named.conf.local

zone "zima.info." {
    type master;
    file "/etc/bind/db.zima.info";
};

zone "wiosna.info." {
    type master;
    file "/etc/bind/db.wiosna.info";
};

us3n00@ks23-3n00:/$

```

Tworzymy i konfigurujemy pliki definiujące nowe strefy. Możemy to zrobić wykorzystując plik definiujący strefę podstawową, **db.szkola.lokalna**

```

sudo cp /etc/bind/db.szkola.lokalna /etc/bind/db.zima.info
sudo cp /etc/bind/db.szkola.lokalna /etc/bind/db.wiosna.info

```

lub utworzonej wcześniej strefy **ferie.info** **db.ferie.info**

```

sudo cp /etc/bind/db.ferie.info /etc/bind/db.zima.info
sudo cp /etc/bind/db.ferie.info /etc/bind/db.wiosna.info

```

Mamy już zdefiniowany adres startowy - rekord SOA oraz serwer nazw - rekord NS. Wystarczy:

- uaktualnić **Zone serial numer**. Numer seryjny ułatwia śledzenia zmian w strefie, przez podrzędny serwer DNS. Zgodnie z konwencją numer seryjny przyjmuje format daty: **yyyymmddss** gdzie **rrrr** to czterocyfrowy numer roku, **mm** to miesiąc, **dd** to dzień a **ss** to numer kolejny w danym dniu. Musimy aktualizować numer seryjny po wprowadzeniu zmian w pliku strefy.
- zdefiniować rekordy hosta nowych stref - rekordy **A**
- zdefiniować aliasy nowych stref - rekordy **CNAME**

W pliku **db.zima.info** :

- w opisie strefy zmieniamy nazwę **szkola.lokalna** na **zima.info**.
- w dyrektywie **\$ORIGIN** zmieniamy nazwę **szkola.lokalna** na **zima.info**.
- w definicji rekordu SOA zmieniamy **serial**, zgodnie z konwencją opisaną powyżej
- w sekcji rekordy **A** zmieniamy adres powiązany z domeną **zima.info** na **172.22.0.5**
- w sekcji rekordy **CNAME** definiujemy alias domeny **zima.info**.. Pamiętajmy, że jeżeli używamy pełnej nazwy domeny, na końcu musi być kropka. Możemy również wykorzystać symbol **@** określający również domenę **zima.info**. - dyrektywa **\$ORIGIN zima.info**.


```

us3n00@ks23-3n00: /
$ cat /etc/bind/db.zima.info
;
; Plik strefy zima.info
;
$TTL      604800; domyslny czas zycia strefy w [s]
$ORIGIN   zima.info.      ;definiowana strefa
;
;bazowe rekordy strefy - kluczowe cechy
@         IN      SOA     ks23-3n00.szkoła.lokalna. root.szkoła.lokalna. (
                                2022030201      ; Serial
                                604800           ; Refresh
                                86400            ; Retry
                                2419200          ; Expire
                                604800 )         ; Negative Cache TTL
;
;serwer nazw
@         IN      NS      ks23-3n00.szkoła.lokalna.
;rekordy A
@         IN      A       172.22.0.5
;
;rekordy CNAME - aliasy
ftp       IN      CNAME   @
www       IN      CNAME   zima.info.
us3n00@ks23-3n00: /$

```

W pliku db.wiosna.info :

- w opisie strefy zmieniamy nazwę szkoła.lokalna .na wiosna.info.
- w dyrektywie \$ORIGIN zmieniamy nazwę szkoła.lokalna. na wiosna.info.
- w definicji rekordu SOA zmieniamy serial, zgodnie z konwencją opisaną powyżej
- w sekcji rekordy A zmieniamy adres powiązany z domeną drzewa.info. na 172.22.0.5
- w sekcji rekordy CNAME definiujemy aliasy domeny wiosna.info.. Pamiętajmy, że jeżeli używamy pełnej nazwy domeny, na końcu musi być kropka. Możemy również wykorzystać symbol @ określający również domenę wiosna.info. - dyrektywa \$ORIGIN wiosna.info.

```

us3n00@ks23-3n00: /
$ cat /etc/bind/db.wiosna.info
;
; Plik strefy wiosna.info
;
$TTL      604800; domyslny czas zycia strefy w [s]
$ORIGIN   wiosna.info.    ;definiowana strefa
;
;bazowe rekordy strefy - kluczowe cechy
@         IN      SOA     ks23-3n00.szkoła.lokalna. root.szkoła.lokalna. (
                                2022030201      ; Serial
                                604800           ; Refresh
                                86400            ; Retry
                                2419200          ; Expire
                                604800 )         ; Negative Cache TTL
;
;serwer nazw
@         IN      NS      ks23-3n00.szkoła.lokalna.
;rekordy A
@         IN      A       172.22.0.5
;
;rekordy CNAME - aliasy
ftp       IN      CNAME   @
www       IN      CNAME   wiosna.info.
us3n00@ks23-3n00: /$

```

Po zapisaniu z modyfikowanych plików, sprawdzamy ich poprawność poleceniami:

sudo named-checkzone zima.info /etc/bind/db.zima.info

sudo named-checkzone wiosna.info /etc/bind/db.wiosna.info

```

us3n00@ks23-3n00: /
us3n00@ks23-3n00:/$ sudo named-checkzone zima.info /etc/bind/db.zima.info
zone zima.info/IN: loaded serial 2022030201
OK
us3n00@ks23-3n00:/$ sudo named-checkzone wiosna.info /etc/bind/db.wiosna.info
zone wiosna.info/IN: loaded serial 2022030201
OK
us3n00@ks23-3n00:/$ █

```

Jeśli nie pamiętamy, czy skonfigurowaliśmy zaporę iptables pod DNS, wpisujemy polecenia.

```

sudo iptables -A INPUT -p tcp --dport 53 -j ACCEPT
sudo iptables -A INPUT -p udp --dport 53 -j ACCEPT

```

Wykonane bezobjawowo świadczy o poprawności.

Możemy teraz restartować usługę bind9 oraz sprawdzić jej status, czy nie ma błędów przy jej uruchamianiu.

```

sudo systemctl restart bind9
sudo systemctl status bind9

```

```

us3n00@ks23-3n00: /
us3n00@ks23-3n00:/$ sudo systemctl restart bind9
us3n00@ks23-3n00:/$ sudo systemctl status bind9
● named.service - BIND Domain Name Server
   Loaded: loaded (/lib/systemd/system/named.service; enabled; vendor preset: enable)
   Active: active (running) since Wed 2022-03-02 11:19:19 UTC; 14s ago
     Docs: man:named(8)
    Main PID: 3681 (named)
      Tasks: 5 (limit: 2268)
     Memory: 14.3M
    CGroup: /system.slice/named.service
            └─3681 /usr/sbin/named -f -u bind

mar 02 11:19:20 ks23-3n00 named[3681]: network unreachable resolving './DNSKEY/IN': 2>
mar 02 11:19:20 ks23-3n00 named[3681]: network unreachable resolving './NS/IN': 2001:>
mar 02 11:19:20 ks23-3n00 named[3681]: network unreachable resolving './DNSKEY/IN': 2>
mar 02 11:19:20 ks23-3n00 named[3681]: network unreachable resolving './NS/IN': 2001:>
mar 02 11:19:20 ks23-3n00 named[3681]: network unreachable resolving './DNSKEY/IN': 2>
mar 02 11:19:20 ks23-3n00 named[3681]: network unreachable resolving './NS/IN': 2001:>
us3n00@ks23-3n00:/$ █

```

Jak widać, wszystkie strefy są załadowane, serwer uruchomiony i nie widać informacji o błędach.

Możemy przetestować działanie serwera. Mamy do tego polecenia **nslookup** (stare działające również w systemach Windows) oraz nowe dające więcej informacji **dig**. Np.:

```

us3n00@ks23-3n00: /
us3n00@ks23-3n00:/$ nslookup www.zima.info
Server:          127.0.0.53
Address:         127.0.0.53#53

Non-authoritative answer:
www.zima.info    canonical name = zima.info.
Name:   zima.info
Address: 172.22.0.5

us3n00@ks23-3n00:/$ █

```

Jeżeli mamy w sieci, skonfigurowanego klienta, możemy wszystkie powyższe testy wykonać na nim.

Np.: **dig www.wiosna.info**

```

ud3n00@k1d23-3n00: /
ud3n00@k1d23-3n00:/$ dig www.wiosna.info

; <<>> DiG 9.16.1-Ubuntu <<>> www.wiosna.info
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 52472
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;www.wiosna.info.                IN      A

;; ANSWER SECTION:
www.wiosna.info.                604800  IN      CNAME   wiosna.info.
wiosna.info.                    7199    IN      A       172.22.0.5

;; Query time: 3 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: śro mar 02 12:25:35 CET 2022
;; MSG SIZE rcvd: 74

ud3n00@k1d23-3n00:/$

```

Teraz możemy przystąpić do konfiguracji serwera Apache2, z witrynami „opartymi na nazwie”. W katalogu domowym użytkownika uxxxy, tworzymy katalog zima, a w nim plik startowy zima.html.

```

us3n00@ks23-3n00: ~/zima
us3n00@ks23-3n00:~/zima$ pwd
/home/us3n00/zima
us3n00@ks23-3n00:~/zima$ cat zima.html
<html>
<head>
  <title> ZIMA </title>
</head>
<body>
  <p> To jest strona powiazana z adresem 172.22.y.5</p>
  <p> domena zima.info</p>
  <p> alias www.zima.info</p>
  <p> To jest strona "oparta na nazwie"</p>
  <p> lokalizacja /home/us3n00/zima/ </p>
  <p> plik startowy zima.html</p>
</body>
</html>
us3n00@ks23-3n00:~/zima$

```

W katalogu głównym, tworzymy katalog wiosna, a w nim plik startowy wiosna.html.

```

us3n00@ks23-3n00: /wiosna
us3n00@ks23-3n00:/wiosna$ cat wiosna.html
<html>
<head>
  <title> WIOSNA </title>
</head>
<body>
  <p> To jest strona powiazana z adresem 172.22.y.5</p>
  <p> domena wiosna.info</p>
  <p> alias www.wiosna.info</p>
  <p> To jest strona "oparta na nazwie"</p>
  <p> lokalizacja /wiosna/ </p>
  <p> plik startowy wiosna.html</p>
</body>
</html>
us3n00@ks23-3n00:/wiosna$

```

Konfigurujemy zasady dostępu do katalogów, gdzie znajdują się katalogi witryn:

/home/uxxyy/zima/, /wiosna/,

```

Otwórz  [icon]  apache2.conf [Tylko do odczytu]  Zapisz  [icon]  [icon]  [icon]
/etc/apache2

170 <Directory /var/www/>
171     Options Indexes FollowSymLinks
172     AllowOverride None
173     Require all granted
174 </Directory>
175
176 <Directory /wiosna/>
177     Options Indexes FollowSymLinks
178     AllowOverride None
179     Require all granted
180 </Directory>
181
182 <Directory /home/us3n00/zima/>
183     Options Indexes FollowSymLinks
184     AllowOverride None
185     Require all granted
186 </Directory>
187

Zwykły tekst  Szerokość tabulacji: 4  Wrsz 176, kol 18  WST

```

Tworzymy w sites-available, hosta wirtualnego zima.conf

```

[icon]  us3n00@ks23-3n00: ~  [icon]  [icon]  [icon]  [icon]

us3n00@ks23-3n00:~$ cat /etc/apache2/sites-available/zima.conf
<VirtualHost 172.22.0.5:80>
    ServerName zima.info
    ServerAlias www.zima.info
    ServerAdmin webmaster@localhost
    DocumentRoot /home/u3n00/zima/
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
us3n00@ks23-3n00:~$

```

Utworzono w sites-available, hosta wirtualnego wiosna.conf

```

[icon]  us3n00@ks23-3n00: ~  [icon]  [icon]  [icon]  [icon]

us3n00@ks23-3n00:~$ cat /etc/apache2/sites-available/wiosna.conf
<VirtualHost 172.22.0.5:80>
    ServerName wiosna.info
    ServerAlias www.wiosna.info
    ServerAdmin webmaster@localhost
    DocumentRoot /wiosna/
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
us3n00@ks23-3n00:~$

```

Sprawdzamy konfigurację portów nasłuchu

```

[icon]  us3n00@ks23-3n00: ~  [icon]  [icon]  [icon]  [icon]

us3n00@ks23-3n00:~$ cat /etc/apache2/ports.conf
# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 80

<IfModule ssl_module>
    Listen 443
</IfModule>

<IfModule mod_gnutls.c>
    Listen 443
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
us3n00@ks23-3n00:~$

```

Dodajemy nowe pliki startowe witryn: wiosna.html, zima.html do pliku dir.conf.

```
us3n00@ks23-3n00: ~  
us3n00@ks23-3n00:~$ cat /etc/apache2/mods-available/dir.conf  
<IfModule mod_dir.c>  
    DirectoryIndex zima.html wiosna.html drzewa.html index.html index.cgi index.pl index.ph  
p index.xhtml index.htm  
</IfModule>  
  
# vim: syntax=apache ts=4 sw=4 sts=4 sr noet  
us3n00@ks23-3n00:~$
```

Włączamy wirtualne hosty zima.conf, wiosna.conf poleceniem a2ensite. (tworzymy dowiązania symboliczne do wymienionych plików w katalogu sites-enabled)

```
us3n00@ks23-3n00:~$ sudo a2ensite zima.conf  
Enabling site zima.  
To activate the new configuration, you need to run:  
    systemctl reload apache2  
us3n00@ks23-3n00:~$ sudo a2ensite wiosna.conf  
Enabling site wiosna.  
To activate the new configuration, you need to run:  
    systemctl reload apache2  
us3n00@ks23-3n00:~$
```

Przeładowano serwer - **service apache2 reload** i sprawdzono jego status - **service apache2 status**.

```
us3n00@ks23-3n00:~$ sudo service apache2 reload  
us3n00@ks23-3n00:~$ sudo service apache2 status  
● apache2.service - The Apache HTTP Server  
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor prese>  
   Active: active (running) since Wed 2022-03-02 09:50:44 UTC; 4h 26min ago  
     Docs: https://httpd.apache.org/docs/2.4/  
  Process: 4562 ExecReload=/usr/sbin/apachectl graceful (code=exited, status=>  
 Main PID: 1036 (apache2)  
    Tasks: 55 (limit: 2268)  
   Memory: 7.4M  
us3n00@ks23-3n00:~$
```

Jak widać nie ma błędów, więc serwis powinien działać poprawnie.

Sprawdzono działanie witryn na kliencie.



