Konfiguracja serwera DNS w systemie Linux - dystrybucja Ubuntu

Wprowadzenie

Ważną częścią zarządzania konfiguracją i infrastrukturą serwera jest utrzymywanie w łatwy sposób wyszukiwania interfejsów sieciowych i adresów IP według nazwy, poprzez skonfigurowanie odpowiedniego pliku System nazw domen (DNS). Używanie w pełni kwalifikowanych nazw domen (FQDN) zamiast IP sieci prywatnej to świetny sposób na usprawnienie zarządzania serwerami.

Poniżej zostaną przedstawione kroki, pokazujące jak skonfigurować wewnętrzny serwer DNS, używając serwera BIND oprogramowanie serwerowe (BIND9) na Ubuntu 16.04 (lub 14.04), z którego mogą korzystać wirtualne serwery prywatne (VPS) do rozwiązywania prywatnych nazw hostów i prywatnych adresów IP.

Zapewnia to centralny sposób zarządzania swoimi wewnętrznymi nazwami hostów i prywatnymi adresami IP, co jest wysoce pożądane, gdy środowisko sieciowe rozszerza się na więcej niż kilka hostów.

Wymagania wstępne

- kilka maszyn, które działają w tym samym centrum danych i w tej samej sieci prywatnej,
- serwer VPS (Virtual Private Server) służący jako podstawowy serwer DNS, ns1,
- opcjonalnie: serwer VPS służący jako pomocniczy serwer DNS, ns2,
- dostęp do konta root

Przykładowe konfiguracja hostów

Do celów testowych, zakładamy następującą konfigurację:

- Mamy dwa istniejące VPS o nazwach „host1” i „host2”
- Oba VPS istnieją w centrum danych „my”
- Oba VPS mają włączoną prywatną sieć (i znajdują się w podsieci 10.128.0.0/16)
- Oba serwery VPS są w jakiś sposób powiązane z aplikacją webową działającą pod adresem „example.com”

Przyjmując te założenia, zdecydowaliśmy, że sensowne jest użycie schematu nazewnictwa, w którym "my.example.com" odnosi się do naszej prywatnej podsieci lub strefy. Dlatego prywatna w pełni kwalifikowana nazwa domeny (FQDN) hosta1 będzie miała postać "host1.my.example.com".

W tabeli poniżej podano dane szczegółowe omawianej konfiguracji:

Host	Rola	FQDN	Adres prywatny IP
host1	ogólny host 1	host1.my.example.com	10.128.100.101
host2	ogólny host 2	host2.my.example.com	10.128.200.102

Uwaga: Docelowa konfiguracja użytkownika będzie inna, ale przykładowe nazwy i adresy IP zostaną użyte do zademonstrowania, jak skonfigurować serwer DNS, aby zapewnić działający wewnętrzny DNS. Należy dostosować przedstawioną konfigurację do własnego środowiska, zastępując nazwy hostów i prywatne adresy IP własnymi. Nie jest konieczne używanie nazwy regionu centrum danych w schemacie nazewnictwa, ale używamy go tutaj, aby wskazać, że te hosty należą do prywatnej sieci określonego centrum danych. Jeśli korzystasz z wielu centrów danych, możesz skonfigurować wewnętrzny DNS w każdym odpowiednim centrum danych.

Nasz cel

Chcemy skonfigurować dwa serwery DNS, podstawowy serwer ns1 i opcjonalnie dodatkowy serwer ns2, który będzie służył jako kopia zapasowa.

Oto tabela z przykładowymi nazwami i adresami IP:

Host	Rola	FQDN	Adres prywatny IP
ns1	Podstawowy serwer DNS	ns1.my.example.com	10.128.10.11
ns2	Zapasowy serwer DNS	ns2.my.example.com	10.128.20.12

Instalacja BIND na serwerach DNS

Uwaga: tekst wyróżniony na czerwono jest ważny! Będzie często używany do oznaczenia czegoś, co należy zastąpić własnymi ustawieniami lub zmodyfikować lub dodać do pliku konfiguracyjnego. Na przykład, jeśli zobaczymy coś takiego jak host1.my.example.com, to należy to zamienić na FQDN własnego serwera. Podobnie, jeśli mamy adres host1_private_IP, należy do zastąpić prywatnym adresem IP własnego serwera.

Na obu serwerach DNS, ns1 i ns2, zaktualizujemy informacje o dostępnych pakietach:

```
sudo apt-get update
```

a następnie instalujemy program bind:

```
sudo apt-get install bind9 bind9utils bind9-doc
```

Tryb IPv4

Zanim przejdziemy dalej, ustawmy BIND na tryb IPv4. Na obu serwerach edytujemy plik parametrów usługi bind9 "etc/default/bind9"

```
sudo vi /etc/default/bind9
```

Dodajemy "-4 -u bind" do zmiennej OPTIONS. Wiersz powinien wyglądać następująco:

```
OPTIONS="-4 -u bind"
```

Zapisujemy i wychodzimy.

Po zainstalowaniu BIND skonfigurujmy podstawowy serwer DNS.

Konfiguracja podstawowego serwera DNS

Konfiguracja BIND składa się z wielu plików, które są zawarte w głównym pliku konfiguracyjnym o nazwie named.conf. Te nazwy plików zaczynają się od „named”, ponieważ jest to nazwa procesu uruchamianego przez program BIND. Zaczniemy od skonfigurowania pliku opcji.

Konfiguracja pliku opcji

Na ns1 otwórzmy plik named.conf.options do edycji:

sudo vi /etc/bind/named.conf.options

```
sudo vi /etc/bind/named.conf.options
```

Nad istniejącym blokiem „options” utwórzmy nowy blok ACL o nazwie „trusted”. W tym miejscu zdefiniujemy listę klientów, z których będziemy zezwalać na rekursywne zapytania DNS (tj. Nasze serwery, które znajdują się w tym samym centrum danych co ns1). Korzystając z naszego przykładowego prywatnego adresu IP, dodamy ns1, ns2, host1 i host2 do naszej listy zaufanych klientów:

```
acl "trusted" {
        10.128.10.11;    # ns1 - can be set to localhost
        10.128.20.12;    # ns2
        10.128.100.101;  # host1
        10.128.200.102;  # host2
};
```

Teraz, gdy mamy naszą listę zaufanych klientów DNS, będziemy chcieli edytować blok opcji. Obecnie początek bloku wygląda następująco:

```
options {
        directory "/var/cache/bind";
...
}
```

Pod dyrektywą „directory” dodaj podświetlone linie konfiguracyjne (i podstawiamy je w odpowiednim adresie IP ns1), aby wyglądało to mniej więcej tak:

```
options {
        directory "/var/cache/bind";

        recursion yes;                 # enables resursive queries
        allow-recursion { trusted; };  # allows recursive queries from
"trusted" clients
        listen-on { 10.128.10.11; };   # ns1 private IP address - listen on
private network only
        allow-transfer { none; };      # disable zone transfers by default

        forwarders {
                8.8.8.8;
                8.8.4.4;
        };
...
};
```

Teraz zapisz i zamknij named.conf.options. Powyższa konfiguracja określa, że tylko Twoje własne serwery (te „zaufane”) będą mogły wysyłać zapytania do serwera DNS.

Następnie skonfigurujemy plik lokalny, aby określić nasze strefy DNS.

Konfiguracja plików lokalnych

Na ns1 otwórzmy plik named.conf.local do edycji:

```
sudo vi /etc/bind/named.conf.local
```

Oprócz kilku komentarzy plik powinien być pusty. Tutaj określimy nasze strefy forward i reverse. Dodaj strefę forward z następującymi wierszami (zastąp nazwę strefy własną):

```
zone "my.example.com" {
    type master;
    file "/etc/bind/zones/db.my.example.com"; # zone file path
    allow-transfer { 10.128.20.12; };         # ns2 private IP address -
secondary
};
```

Zakładając, że nasza prywatna podsieć to 10.128.0.0/16, dodajemy strefę odwrotną za pomocą następujących wierszy (zwróćmy uwagę, że nazwa naszej strefy odwrotnej zaczyna się od „128.10”, co jest odwróceniem oktetu „10.128”):

```
zone "128.10.in-addr.arpa" {
    type master;
    file "/etc/bind/zones/db.10.128";  # 10.128.0.0/16 subnet
    allow-transfer { 10.128.20.12; };  # ns2 private IP address - secondary
};
```

Jeśli Twoje serwery obejmują wiele prywatnych podsieci, ale znajdują się w tym samym centrum danych, pamiętaj o określeniu dodatkowej strefy i pliku strefy dla każdej odrębnej podsieci. Po zakończeniu dodawania wszystkich żądanych stref zapisz i zamknij plik named.conf.local.

Teraz, gdy nasze strefy są określone w BIND, musimy utworzyć odpowiednie pliki stref do przodu i do tyłu.

Tworzenie pliku strefy do przodu

Plik strefy do przodu to miejsce, w którym definiujemy rekordy DNS na potrzeby wyszukiwania DNS do przodu. Oznacza to, ze gdy serwer DNS otrzyma zapytanie o nazwę, na przykład „host1.my.example.com", będzie szukał w pliku strefy do przodu, aby znaleźć odpowiedni prywatny adres IP hosta1.

Utwórzmy katalog, w którym będą znajdować się nasze pliki stref. Zgodnie z nasza konfiguracja named.conf.local tą lokalizacja powinno być `/etc/bind/zones:`

```
sudo mkdir /etc/bind/zones
```

Nasz plik forward zone opieramy na przykładowym pliku db.local zone. Skopiujmy go do odpowiedniej lokalizacji za pomocą następujących poleceń:

```
cd /etc/bind/zones
sudo cp ../db.local ./db.my.example.com
```

Teraz edytujmy nasz plik strefy do przodu:

```
sudo vi /etc/bind/zones/db.my.example.com
```

Początkowo będzie wyglądać mniej więcej tak:

```
$TTL    604800
@       IN      SOA     localhost. root.localhost. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      localhost.      ; delete this line
```

```
@       IN      A       127.0.0.1       ; delete this line
@       IN      AAAA    ::1             ; delete this line
```

Najpierw będziemy chcieli edytować rekord SOA. Zastąpmy pierwszy „localhost” nazwą FQDN ns1, a następnie „root.localhost” zastąpmy „admin.my.example.com”. Ponadto za każdym razem, gdy edytujemy plik strefy, przed ponownym uruchomieniem nazwanego procesu należy zwiększyć wartość seryjną - zwiększymy ją do „3”. Wpis powinien wyglądać mniej więcej tak:

```
@       IN      SOA     ns1.my.example.com. admin.my.example.com. (
                              3         ; Serial
```

Teraz usuńmy trzy rekordy na końcu pliku (po rekordzie SOA). Jeśli nie mamy pewności, które linie usunąć, są one oznaczone powyższym komentarzem „delete this line”.

Na końcu pliku dodajmy rekordy serwera nazw następującymi wierszami (zastąpmy nazwy własnymi). Zwróćmy uwagę, że druga kolumna określa, że są to rekordy „NS”:

```
; name servers - NS records
    IN      NS      ns1.my.example.com.
    IN      NS      ns2.my.example.com.
```

Następnie dodaj rekordy A dla swoich hostów należących do tej strefy. Obejmuje to każdy serwer, którego nazwa ma kończyć się ciągiem „.my.example.com” (należy zastąpić nazwy i prywatne adresy IP). Korzystając z naszych przykładowych nazw i prywatnych adresów IP, dodamy rekordy A dla ns1, ns2, host1 i host2 w następujący sposób:

```
; name servers - A records
ns1.my.example.com.          IN      A       10.128.10.11
ns2.my.example.com.          IN      A       10.128.20.12

; 10.128.0.0/16 - A records
host1.my.example.com.        IN      A      10.128.100.101
host2.my.example.com.        IN      A      10.128.200.102
```

Zapisujemy i zamykamy plik db.my.example.com.

Nasz ostatni przykładowy plik strefy do przodu wygląda następująco:

```
$TTL    604800
@       IN      SOA     ns1.my.example.com. admin.my.example.com. (
                  3       ; Serial
             604800     ; Refresh
              86400     ; Retry
            2419200     ; Expire
             604800 )   ; Negative Cache TTL
;
; name servers - NS records
     IN      NS      ns1.my.example.com.
     IN      NS      ns2.my.example.com.

; name servers - A records
```

```
ns1.my.example.com.           IN      A       10.128.10.11
ns2.my.example.com.           IN      A       10.128.20.12

; 10.128.0.0/16 - A records
host1.my.example.com.         IN      A      10.128.100.101
host2.my.example.com.         IN      A      10.128.200.102
```

Teraz przejdźmy do pliku (ów) strefy odwrotnej.

Plik strefy odwróconej

Plik strefy odwrotnej to miejsce, w którym definiujemy rekordy DNS PTR na potrzeby odwrotnego wyszukiwania DNS. Oznacza to, że gdy serwer DNS otrzyma zapytanie na podstawie adresu IP, na przykład „10.128.100.101", zajrzy do plików strefy odwrotnej, aby rozpoznać odpowiednią nazwę FQDN, w tym przypadku „host1.my.example.com" .

Na ns1 dla każdej strefy odwróconej określonej w pliku named.conf.local utwórzmy plik strefy odwróconej. Nasz plik (i) strefy odwrotnej opieramy na przykładowym pliku strefy db.127. Skopiujmy go do odpowiedniej lokalizacji za pomocą następujących poleceń (zastępując nazwę pliku docelowego, aby pasowała do definicji strefy odwrotnej):

```
cd /etc/bind/zones

sudo cp ../db.127 ./db.10.128
```

Edytujemy plik odwróconej strefy, który odpowiada odwróconej strefie (strefom) zdefiniowanym w named.conf.local:

```
sudo vi /etc/bind/zones/db.10.128
```

Początkowo będzie wyglądać mniej więcej tak:

```
$TTL    604800
@       IN      SOA     localhost. root.localhost. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      localhost.      ; delete this line
1.0.0   IN      PTR     localhost.      ; delete this line
```

W taki sam sposób, jak w przypadku pliku strefy do przodu, będziemy edytować rekord SOA i zwiększać wartość seryjną. Powinien wyglądać mniej więcej tak:

```
@       IN      SOA     ns1.my.example.com. admin.my.example.com. (
```

```
                                3          ; Serial
```

Teraz usuńmy dwa rekordy na końcu pliku (po rekordzie SOA). Jeśli nie mamy pewności, które linie usunąć, są one oznaczone powyższym komentarzem „usuń tę linię”.

Na końcu pliku dodajmy rekordy serwera nazw następującymi wierszami (należy zastąpić nazwy własnymi). Zwróćmy uwagę, że druga kolumna określa, że są to rekordy „NS”:

```
; name servers - NS records
      IN      NS      ns1.my.example.com.
      IN      NS      ns2.my.example.com.
```

Następnie dodajemy rekordy PTR dla wszystkich serwerów, których adresy IP znajdują się w podsieci pliku strefy, który edytujesz. W naszym przykładzie obejmuje to wszystkie nasze hosty, ponieważ wszystkie znajdują się w podsieci 10.128.0.0/16. Zwróćmy uwagę, że pierwsza kolumna składa się z dwóch ostatnich oktetów prywatnych adresów IP serwerów w odwrotnej kolejności. Pamiętajmy, aby zastąpić nazwy i prywatne adresy IP, aby pasowały do naszych serwerów:

```
; PTR Records
11.10   IN      PTR     ns1.my.example.com.    ; 10.128.10.11
12.20   IN      PTR     ns2.my.example.com.    ; 10.128.20.12
101.100 IN      PTR     host1.my.example.com.  ; 10.128.100.101
102.200 IN      PTR     host2.my.example.com.  ; 10.128.200.102
```

Zapisz i zamknij plik strefy odwróconej (powtórz tę sekcję, jeśli chcesz dodać więcej plików strefy odwróconej).

Nasz ostatni przykładowy plik strefy odwrotnej wygląda następująco:

```
$TTL    604800
@       IN      SOA     my.example.com. admin.my.example.com. (
                              3         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
; name servers
      IN      NS      ns1.my.example.com.
      IN      NS      ns2.my.example.com.

; PTR Records
11.10   IN      PTR     ns1.my.example.com.    ; 10.128.10.11
12.20   IN      PTR     ns2.my.example.com.    ; 10.128.20.12
101.100 IN      PTR     host1.my.example.com.  ; 10.128.100.101
102.200 IN      PTR     host2.my.example.com.  ; 10.128.200.102
```

Sprawdzamy składnię konfiguracji BIND

Uruchom następującą komendę, aby sprawdzić składnię plików named.conf *:

```
sudo named-checkconf
```

Jeśli nazwane pliki konfiguracyjne nie zawierają błędów składniowych, system wróci do znaku zachęty powłoki i nie zobaczymy żadnych komunikatów o błędach. Jeśli występują problemy z plikami konfiguracyjnymi, zapoznajmy się z komunikatem o błędzie. Po korekcie należy ponownie spróbuj wykonać named-checkconf.

Polecenie named-checkzone może być użyte do sprawdzenia poprawności plików stref. Jego pierwszy argument określa nazwę strefy, a drugi argument określa odpowiedni plik strefy, które są zdefiniowane w named.conf.local.

Na przykład, aby sprawdzić konfigurację strefy przekazywania „my.example.com”, należy uruchomić następujące polecenie (zmień nazwy, aby pasowały do strefy przekazywania i pliku):

```
sudo named-checkzone my.example.com db.my.example.com
```

Aby sprawdzić konfigurację odwróconej strefy „128.10.in-addr.arpa”, należy uruchomić następującą komendę (zmień numery, aby pasowały do odwróconej strefy i pliku):

```
sudo named-checkzone 128.10.in-addr.arpa /etc/bind/zones/db.10.128
```

Gdy wszystkie pliki konfiguracyjne i pliki stref nie zawierają błędów, możemy ponownie uruchomić usługę BIND.

Ponowne uruchomienie serwisu BIND

```
sudo service bind9 restart
```

Nasz podstawowy serwer DNS jest teraz skonfigurowany i gotowy do odpowiadania na zapytania DNS. Przejdźmy do tworzenia pomocniczego serwera DNS.

Konfiguracja pomocniczego serwera DNS

W większości środowisk dobrym pomysłem jest skonfigurowanie pomocniczego serwera DNS, który będzie odpowiadał na żądania, jeśli podstawowy będzie niedostępny. Na szczęście pomocniczy serwer DNS jest znacznie łatwiejszy do skonfigurowania.

Na ns2 edytujemy plik named.conf.options:

```
sudo vi /etc/bind/named.conf.options
```

U góry pliku dodajemy listę ACL z prywatnymi adresami IP wszystkich zaufanych serwerów:

```
acl "trusted" {
        10.128.10.11;    # ns1
        10.128.20.12;    # ns2 - can be set to localhost
        10.128.100.101;  # host1
        10.128.200.102;  # host2
};
```

Pod dyrektywą katalogu dodajmy następujące wiersze:

```
recursion yes;
        allow-recursion { trusted; };
        listen-on { 10.128.20.12; };      # ns2 private IP address
        allow-transfer { none; };          # disable zone transfers by
default

        forwarders {
                8.8.8.8;
                8.8.4.4;
        };
```

Zapisujemy i zamykamy named.conf.options. Ten plik powinien wyglądać dokładnie tak, jak plik named.conf.options z ns1, z wyjątkiem tego, że powinien być skonfigurowany do nasłuchiwania na prywatnym adresie IP ns2.

Teraz edytujemy plik named.conf.local:

```
sudo vi /etc/bind/named.conf.local
```

Zdefiniujmy strefy podrzędne, które odpowiadają strefom głównym na podstawowym serwerze DNS. Zauważmy, że typ to „slave”, plik nie zawiera ścieżki i istnieje dyrektywa master, która powinna być ustawiona na prywatny adres IP podstawowego serwera DNS. Jeśli zdefiniowaliśmy wiele stref odwrotnych na podstawowym serwerze DNS, pamiętajmy, aby dodać je wszystkie tutaj:

```
zone "my.example.com" {
    type slave;
    file "db.my.example.com";
    masters { 10.128.10.11; };  # ns1 private IP
};

zone "128.10.in-addr.arpa" {
    type slave;
    file "db.10.128";
```

```
    masters { 10.128.10.11; };  # ns1 private IP
};
```

Teraz należy zapisać i zamknąć plik named.conf.local.

Uruchom następujące polecenie, aby sprawdzić poprawność plików konfiguracyjnych:

```
sudo named-checkconf
```

Po sprawdzeniu uruchom ponownie bind:

```
sudo service bind9 restart
```

Na podstawie przedstawionej konfiguracji mamy podstawowe i pomocnicze serwery DNS do rozpoznawania nazw sieci prywatnych i adresów IP. Teraz możemy skonfigurować komputery w sieci lokalnej do korzystania z prywatnych serwerów DNS.

Konfiguracja klientów DNS

Zanim wszystkie nasze serwery na „zaufanej" liście ACL będą mogły wysyłać zapytania do naszych serwerów DNS, musimy skonfigurować każdy z nich tak, aby używał ns1 i ns2 jako serwerów nazw. Ten proces różni się w zależności od systemu operacyjnego, ale w przypadku większości dystrybucji Linuksa wymaga dodania serwerów nazw do pliku /etc/resolv.conf.

Klienci Ubuntu

W systemie Ubuntu i Debian Linux VPS możesz edytować plik head, który jest dołączany do resolv.conf podczas rozruchu:

```
sudo vi /etc/resolvconf/resolv.conf.d/head
```

Dodajmy następujące wiersze do pliku (zastąp swoją domenę prywatną oraz prywatne adresy IP ns1 i ns2):

```
search my.example.com  # your private domain
nameserver 10.128.10.11  # ns1 private IP address
nameserver 10.128.20.12  # ns2 private IP address
```

Teraz należy uruchomić resolvconf, aby wygenerować nowy plik resolv.conf:

```
sudo resolvconf -u
```

Mamy teraz skonfigurowane hosty, aby korzystały z lokalnych serwerów DNS.