Bài viết này giới thiệu với các bạn về cài đặt và cấu hình DNS Server của hệ điều hành Linux. Trong bài viết này sẽ giới thiệu với các bạn các loại dns server cùng với cách cài đặt dns trên bản phân phối CentOS 7 từ đó giúp cho chúng ta có thể tự tìm hiểu và tự học Linux cơ bản dễ dàng hơn.

1. Giới thiệu về DNS, DNS Server và các loại DNS Server

**1.1. Giới thiệu về DNS**

DNS(Domain Name System) là một hệ thống dùng để phân giải tên miền. Nó dùng để thiết lập mối quan hệ giữa tên tên miền và địa chỉ IP. DNS giúp chúng ta có thể chuyển đổi tên miền (blogd.net) sang một địa chỉ IP (192.168.1.15).

Ví dụ: Khi chúng ta vào google và gõ vào thanh tìm kiếm site "blogd.net" thì ngay lập tức DNS sẽ phân giải và trả về kết quả với IP là 192.168.1.15.

**1.2. Giới thiệu về DNS Server**

DNS Server là một loại máy chủ dùng để quản lý và xử lý các tên miền thực hiện các bản ghi liên quan của chúng. DNS Server là thành phần chính thực hiện giao thức DNS và cung cấp các dịch vụ phân giải tên miền cho máy chủ và máy khách web trên mạng dựa trên IP. DNS Server dùng để định vị và phân phối các trang web cho người dùng cuối qua Internet, máy chủ DNS được phát triển trên phần cứng thông thường nhưng chạy phần mềm DNS chuyên dụng. Nó luôn được kết nối với Internet.

DNS Server dùng để chứa cơ sở dữ liệu bao gồm các địa chỉ IP ứng với tên miền nhất định được gọi là bản ghi DNS. Nó thực hiện chức năng cơ bản nhất là phân giải tên miền thành địa chỉ IP tương ứng. Trong quá trình phân giải tên miền thì kết quả tìm kiếm nếu có trong các bản ghi DNS thì tên miền được trả về ứng với địa chỉ IP của nó. Nếu tên miền không được đăng ký hoặc thêm vào máy chủ DNS đó, truy vấn sẽ được chuyển đến các máy chủ DNS khác cho đến khi tìm thấy bản ghi tên miền.

Phần mềm DNS Server nổi tiếng nhất là BIND hoàn toàn miễn phí và được phân phối với các hệ thống Linux. Đối với các hệ thống sử dụng Microsoft thì Microsoft DNS là một phần mềm phổ biến của Windows Server.

**1.3. Các loại DNS Server**

Tất cả các máy chủ DNS thuộc một trong 4 loại DNS Server sau:

* Recursive resolver
* Root nameserver
* TLD nameserver
* Authoritative nameserver

Khi chúng ta thực hiện tra cứu DNS thì các máy chủ này sẽ phối hợp với nhau để hoàn thành nhiệm vụ cung cấp IP cho tên miền mà chúng ta tìm kiếm.

**1.3.1. Giới thiệu về recursive resolver**

Một recursive resolver là điểm dừng đầu tiên trong quá trình truy vấn DNS. Nó hoạt động như một cầu nối trung gian giữa máy client và DNS nameserver.

Sau khi nhận được truy vấn DNS từ một client web thì recursive resolver sẽ phản hồi với dữ liệu được lưu trong bộ nhớ cache hoặc gửi yêu cầu đến root nameservers tiếp theo là đến TLD nameservers và cuối cùng đến authoritative nameserver.

Sau khi nhận được phản hồi từ authoritative nameserver chứa địa chỉ IP được yêu cầu, recursive resolver sẽ gửi phản hồi cho Client.

Trong quá trình này, recursive resolver sẽ lưu trữ thông tin nhận được từ authoritative nameserver. Khi một Client yêu cầu địa chỉ IP của một tên miền giống với client trước yêu cầu thì recursive resolver sẽ cung cấp các bản ghi được yêu cầu từ bộ nhớ cache của nó. Bỏ qua quá trình hỏi root nameservers.

**1.3.2. Giới thiệu về Root nameservers**

Trên thế giới hiện tại có 13 Root nameservers DNS. Các Root nameservers DNS này sẽ chứa các bản ghi gồm toàn bộ các thông tin về tên miền cùng với địa chỉ IP ứng với tên miền đó. Đây cũng chính là điểm dừng đầu tiên trong quá trình phân giải DNS.

Root nameservers chấp nhận truy vấn của recursive resolver gồm một tên miền và thực hiện trả lời bằng cách hướng recursive resolver đến TLD nameservers dựa trên phần mở rộng của tên miền đó (.com, .vn, .net, .org,...). Các Root nameservers này được giảm sác và theo dõi bỡi Internet Corporation for Assigned Names and Numbers (ICANN).

**Lưu ý:** Có 13 Root nameservers nhưng có nhiều bản sao của mỗi loại trên toàn thế giới sử dụng định tuyến Anycast để cung cấp các trả lời nhanh chóng.

| **Tên của DNS-Server** | **Địa chỉ IPv4** | **Địa chỉ IPv6** | **Nhà điều hành** |
| --- | --- | --- | --- |
| A | 198.41.0.4 | 2001:503:ba3e::2:30 | VeriSign |
| B | 192.228.79.201 | 2001:478:65::53 | USC-ISI |
| C | 192.33.4.12 | 2001:500:2::c | Cogent Communications |
| D | 199.7.91.13 | 2001:500:2d::d | University of Maryland |
| E | 192.203.230.10 |  | NASA |
| F | 192.5.5.241 | 2001:500:2f::f | ISC |
| G | 192.112.36.4 |  | U.S. DoD NIC |
| H | 128.63.2.53 | 2001:500:1::803f:235 | US Army Research Lab |
| I | 192.36.148.17 | 2001:7FE::53 | Autonomica |
| J | 192.58.128.30 | 2001:503:c27::2:30 | VeriSign |
| K | 193.0.14.129 | 2001:7fd::1 | RIPE NCC |
| L | 199.7.83.42 | 2001:500:3::42 | ICANN |
| M | 202.12.27.33 | 2001:dc3::35 | WIDE Project |

**1.3.3. Giới thiệu về TLD nameservers**

TLD nameservers dùng để duy trì thông tin cho tất cả các tên miền có chung một phần mở rộng như chúng ta đã nói bên trên (.com, .vn, .net, .org,...).

Ví dụ: Trên trình duyệt web chúng ta thực hiện gõ vào thanh tìm kiếm google.com khi nhận được phản hồi từ Root nameservers thì recursive resolver sẽ gửi truy vấn đến TLD nameservers. Tại đây chúng ta nhận được câu trả lời chỉ cách chúng ta đến máy chủ tên có thẩm quyền cho tên miền có phần mở rộng là .com .

Việc quản lý các máy TLD nameservers bởi Internet Assigned Numbers Authority (IANA), một chi nhánh của ICANN. IANA chia các máy chủ TLD thành hai nhóm chính:

* Các tên miền cấp cao chung: Đây là các tên miền không cụ thể theo quốc gia, một số TLD chung được biết đến nhiều nhất bao gồm .com, .org, .net, .edu và .gov.
* Tên miền cấp cao nhất của mã quốc gia: Chúng bao gồm bất kỳ tên miền nào dành riêng cho một quốc gia hoặc tiểu bang. Ví dụ bao gồm .vn, .uk, .us, .ru và .jp.

**1.3.4. Giới thiệu về authoritative nameserver**

Khi recursive resolver nhận được phản hồi từ TLD nameservers, phản hồi đó sẽ hướng recursive resolver đến authoritative nameserver. Đây chính là bước cuối cùng mà recursive resolver sẽ tìm được tìm địa chỉ IP cần tìm kiếm.

Authoritative nameserves là nơi chứa thông tin cụ thể của các tên miền như là: google.com và nó có thể cung cấp cho recursive resolver địa chỉ IP của máy chủ đó được tìm thấy trong bản ghi DNS. Nếu như tên miền có bản ghi CNAME (alias) nó sẽ cung cấp cho recursive resolver với một miền alias mà tại đây recursive resolver sẽ phải thực hiện tra cứu DNS hoàn toàn mới để tạo một bản ghi từ authoritative nameserver khác.

**Ví dụ:** Sau khi tìm hiểu khái niệm về Recursive resolver, Root nameserver, TLD nameserver, Authoritative nameserver chúng ta thực hiện xác định 4 loại Server khi truy cập một webste (ví dụ: www.kenh14.vn) chúng ta thực thi lệnh sau:

[root@localhost ~]# dig +trace www.kenh14.vn

; <<>> DiG 9.11.4-P2-RedHat-9.11.4-9.P2.el7 <<>> +trace www.kenh14.vn

;; global options: +cmd

. 86871 IN NS a.root-servers.net.

. 86871 IN NS b.root-servers.net.

. 86871 IN NS c.root-servers.net.

. 86871 IN NS d.root-servers.net.

. 86871 IN NS e.root-servers.net.

. 86871 IN NS f.root-servers.net.

. 86871 IN NS g.root-servers.net.

. 86871 IN NS h.root-servers.net.

. 86871 IN NS i.root-servers.net.

. 86871 IN NS j.root-servers.net.

. 86871 IN NS k.root-servers.net.

. 86871 IN NS l.root-servers.net.

. 86871 IN NS m.root-servers.net.

. 86871 IN RRSIG NS 8 0 518400 20191005170000 20190922160000 59944 . QrWG/QmL8Ia6xVHtrRudwauffTIwNvDAN3s3VNP9ERJS4/+bUVVloU2Y uiuYrZwdKAClp9VFk41uaUeX0aDMD4uEZhbkNte5pC+TD6+BdX00G59s 0ihXn4k7EKZUkVNYGR2SX5WvLvmSNx0jj3Kfygk55a83EZuT0QeZxgwM 7qt6zZfPCO4VClGe3i7K61mP+cDA/t5vHWgTPnQXdUsVE9w1Xz25OlhU DO6tFwvI4tGC1sB+hbixi4IaVRj4EA/HF1WNXMuyRLsBibq0inPlDser d4biK1UzP0IaYn9sjOECpC145b9g8suXGnyRjMMt8YWz/F4A7h8k12Rd 2T56Zg==

;; Received 525 bytes from 8.8.4.4#53(8.8.4.4) in 21 ms

vn. 172800 IN NS a.dns-servers.vn.

vn. 172800 IN NS b.dns-servers.vn.

vn. 172800 IN NS c.dns-servers.vn.

vn. 172800 IN NS d.dns-servers.vn.

vn. 172800 IN NS e.dns-servers.vn.

vn. 172800 IN NS f.dns-servers.vn.

vn. 172800 IN NS g.dns-servers.vn.

vn. 86400 IN DS 3260 8 1 913C652B4006DEAC045B945F4FFCBC1065645421

vn. 86400 IN DS 3260 8 2 5A58C19AF266077FFE16C2668812796FA8193661AF569DBCEB177F91 1B5E6167

vn. 86400 IN RRSIG DS 8 1 86400 20191006050000 20190923040000 59944 . Mi1EEOM60ItP4PYWwSW4j2ngtTcEt4HGZ4ffZFv5uCWoS//3+FioZwCY 1Pl1DBgLDs2Nq6UrA76WWI8QhwhQMQQbLv6xx2fB9mcOvgZRFM5At+cE 8Txd0fBikk0AqTvDP0G/w5Fo5LHX+QmF4rJWYv7Wsf4/FXogs4TyliLt fEtX+6U9DqfwzAJQSDyoeFFAmOcE3muVbF+Pg3uwaHqKI4rDrfUF9LTK BthZH3fStYuJZ7jJlWUALsi9y2G+Fo6U4mYJfrApi3ePAPpwEC6W9iuQ HBt4rm7ca0u0+SWptHmCYhjK+i37z036XNbbMLiQ+N87y62r2ny5Tvat 8xvwuQ==

;; Received 817 bytes from 192.5.5.241#53(f.root-servers.net) in 69 ms

kenh14.vn. 43200 IN NS ns7.synerfy.vn.

kenh14.vn. 43200 IN NS ns8.synerfy.vn.

kenh14.vn. 43200 IN NS ns6.synerfy.vn.

VB8HCP5TJ2C2F9ME3Q5PDEH49JR1UKD1.vn. 5400 IN NSEC3 1 1 5 4B433A97 VFARF57JMG471JFVTKQ6AGHOBIEIFH0I NS SOA RRSIG DNSKEY NSEC3PARAM

VB8HCP5TJ2C2F9ME3Q5PDEH49JR1UKD1.vn. 5400 IN RRSIG NSEC3 8 2 5400 20191018101306 20190918092133 21051 vn. EdE5F7X/+0ov07j2jsfFDVSqHRFnQWbl9k+GpCNybDa+BVy3vHGj7+tj FMIxWO3rsS8zzYSo6JEAJetVM0qR30oePT5SCk8R4MS4fdM8i1EqnBwW xj9HQnlTGGcpiJyfNsRq3+/iil6DWmG/6qhvhCiRib2EN+mT8QHsMODZ 6eU=

SA0CSOFPRD5M19IH9KVL1PI1LCPSB9U8.vn. 5400 IN NSEC3 1 1 5 4B433A97 SD1VP058UMTUPH5R4NH9GPTU90CT783L NS DS RRSIG

SA0CSOFPRD5M19IH9KVL1PI1LCPSB9U8.vn. 5400 IN RRSIG NSEC3 8 2 5400 20191001074308 20190901072034 21051 vn. bvGOVgJhaSBJgv6DfkZObfEOicrCAT3ya3C2Rvx63FfYlGaC4txRzk3q DdgQeLSOapCNYnYyBP6zJryZcjNXUAebfgNvpcRwyqhzrcLMv9kgG/fR 1Chnz8Kf/AJICKKadvHJ+p7U0/+8goUHv8nOJ/g8zRHANbXof/ZWYg4V /NI=

;; Received 595 bytes from 203.119.68.105#53(f.dns-servers.vn) in 10 ms

www.kenh14.vn. 3600 IN A 222.255.27.165

kenh14.vn. 3600 IN NS ns6.synerfy.vn.

kenh14.vn. 3600 IN NS ns7.synerfy.vn.

kenh14.vn. 3600 IN NS ns8.synerfy.vn.

;; Received 168 bytes from 173.193.24.114#53(ns6.synerfy.vn) in 208 ms

Qua kết quả trên chúng ta thấy khi truy vấn một URL từ trình duyệt web thì chúng sẽ thực hiện các bước sau:

* Bước 1: Khi truy vấn URL (www.kenh14.vn) trên Client thì recursive resolver sẽ là cầu nối trung gian giữa máy client và DNS nameserver.

Theo kết quả trên:

;; Received 525 bytes from 8.8.4.4#53(8.8.4.4) in 21 ms

* Bước 2: Recursive resolver thực hiện truy vấn Root nameserver cho (www.kenh14.vn).

Chúng ta được kết quả sau:

;; Received 817 bytes from 192.5.5.241#53(f.root-servers.net) in 69 ms

* Bước 3: Root nameserver sẽ chỉ dẫn recursive resolver của chúng ta đến TLD nameserver của tên miền .vn.

Kết quả tìm kiếm TLD nameserver:

;; Received 595 bytes from 203.119.68.105#53(f.dns-servers.vn) in 10 ms

* Bước 4: Recursive resolver của chúng ta truy vấn lên TLD nameserver .vn tiếp theo nó sẽ được giới thiệu tới Authoritative nameserver nơi chứa thông tin cụ thể của tên miền (www.kenh14.vn).

Kết quả xác định Authoritative nameserver:

;; Received 168 bytes from 173.193.24.114#53(ns6.synerfy.vn) in 208 ms

Sau khi recursive resolver truy vấn các Authoritative nameserver cho www.kenh14.vn và nhận câu trả lời thì recursive resolver sẽ lưu trữ câu trả lời trong khoảng thời gian tồn tại (TTL) được chỉ định trong cấu hình của chúng ta.

Đây là chính quá trình diễn ra sau khi chúng ta nhập một URL vào trình duyệt web.

**1.4. Các loại Bản ghi DNS**

**1.4.1. Khái Niệm DNS Resource Record**

* **Domain Name System DNS**: là một cơ sở dữ liệu phân tán, phân cấp, và dự phòng cho các thông tin liên quan đến tên miền, địa chỉ Internet. Trong các máy chủ tên miền, các loại record khác nhau được sử dụng cho mục đích khác nhau.
* **Resource Record (RR)** : là mẫu thông tin dùng để mô tả các thông tin về cơ sở dữ liệu DNS, các mẫu thông tin này được lưu trong các file cơ sở dữ liệu của DNS (*%systemroot%system32dns*).
* Có rất nhiều loại RR khác nhau. Khi một zone mới được tạo ra, DNS tự động thêm 2 RR vào zone đó là: **Start of Authority (SOA)** và **Name Server (NS)**.

**1.4.2. Các Kiểu Resource Records Của DNS:**

**1. SOA (Start of Authority)** : Trong mỗi tập tin cơ sở dữ liệu DNS phải có một và chỉ một record SOA (Start of Authority). Bao gồm các thông tin về domain trên DNS Server, thông tin về zone transfer.

* **Cú pháp :**

[tên miền] IN SOA [tên-server-dns] [địa-chỉ-email] (serial number;refresh number;retry number;experi number;time-to-live number)

* *Serial* : áp dụng cho mọi dữ liệu trong zone và là một số nguyên. Định dạng này theo kiểu YYYYMMDDNN, trong đó YYYY là năm, MM là tháng, DD là ngày, NN là số lần sửa đổi dữ liệu zone trong ngày. Bất kể là theo định dạng nào, luôn luôn phải tăng số này lên mỗi lần sửa đổi dữ liệu zone. Khi máy chủ Secondary liên lạc với máy chủ Primary, trước tiên nó sẽ hỏi số serial. Nếu số serial của máy Secondary nhỏ hơn số serial của máy Primary tức là dữ liệu zone trên Secondary đã cũ và sao đó máy Secondary sẽ sao chép dữ liệu mới từ máy Primary thay cho dữ liệu đang có.
* *Refresh* : chỉ ra khoảng thời gian máy chủ Secondary kiểm tra sữ liệu zone trên máy Primary để cập nhật nếu cần. Giá trị này thay đổi tùy theo tuần suất thay đổi dữ liệu trong zone.
* *Retry* : nếu máy chủ Secondary không kết nối được với máy chủ Primary theo thời hạn mô tả trong refresh (ví dụ máy chủ Primary bị shutdown vào lúc đó thì máy chủ Secondary phải tìm cách kết nối lại với máy chủ Primary theo một chu kỳ thời gian mô tả trong retry. Thông thường, giá trị này nhỏ hơn giá trị refresh).
* *Expire* : nếu sau khoảng thời gian này mà máy chủ Secondary không kết nối được với máy chủ Primary thì dữ liệu zone trên máy Secondary sẽ bị quá hạn. Một khi dữ liệu trên Secondary bị quá hạn thì máy chủ này sẽ không trả lời mỗi truy vấn về zone này nữa. Giá trị expire này phải lớn hơn giá trị refresh và giá trị retry.
* *TTL (time to live)* : giá trị này áp dụng cho mọi record trong zone và được đính kèm trong thông tin trả lời một truy vấn. Mục đích của nó là chỉ ra thời gian mà các máy chủ name server khác cache lại thông tin trả lời. Việc cache thông tin trả lời giúp giảm lưu lượng truy vấn DNS trên mạng.

**2. NS (Name Server)** : Record tiếp theo cần có trong zone là NS (name server) record. Mỗi name server cho zone sẽ có một NS record. Chứa địa chỉ IP của DNS Server cùng với các thông tin về domain đó.

* **Cú pháp :**

[domain\_name] IN NS [DNS-Server\_name]

* Ví dụ : Record NS sau :  
  Matbao.com. IN NS ns1.matbao.com.  
  Matbao.com. IN NS ns2.matbao.com.  
  Chỉ ra hai name servers cho miền matbao.com.

**3. A (Address) và Cname (Canonical Name)** :

* **A Record – Address Record** : dùng để phân giải Host ra một địa chỉ 32-bit IPv4. Dùng để trỏ tên website như www.domain.com đến một Server Hosting website đó.
* **Record CNAME (canonical name)**: tạo tên bí danh (alias) trỏ vào Server Hosting website đó.Thông thường thì máy tính trên Internet có nhiều dịch vụ như Web Server, FTP Server, Chat Server, …. Để lọc hay nói nôm na là kiểm soát, CNAME Records đã được sử dụng.
* Ví dụ :  
  matbao.com. IN CNAME www.matbao.com  
  matbao.com. IN A 112.78.2.100

**4. AAAA** : dùng để phân giải Host ra một địa chỉ 128-bit IPv6.

**5. SRV** : Cung cấp cơ chế định vị dịch vụ, Active Directory sử dụng resource record này để xác định domain controllers, global catalog servers, Lightweight Directory Access Protocol (LDAP) servers. Các trường trong record SVR :

* Tên dịch vụ service.
* Giao thức sử dụng.
* Tên miền (domain name).
* TTL và class.
* Priority.
* weight (hỗ trợ load balancing).
* Port của dịch vụ.
* Target chỉ định FQDN cho host hỗ trợ dịch vụ.

**6. MX (Mail Exchange)** : Dùng để xác định Mail Server cho một domain. Ví dụ khi bạn gởi email tới support@matbao.com, mail server sẽ xem xét MX Record matbao.com xem nó được điểu khiển chính xác bởi mail server nào (mail.matbao.com chẳng hạn) rồi tiếp đến sẽ xem A Record để chuyển tới IP đích. Để tránh việc gởi mail bị lặp lại, record MX có thêm một giá trị bổ sung ngoài tên miền của Mail Exchange là một số thứ tự tham chiếu. Đây là giá trị nguyên không dấu 16-bit (0-65535) chỉ ra thứ tự ưu tiên của các mail exchanger.

* **Cú pháp :**

[domain\_name] IN MX [priority] [mail-host]

* Ví dụ : matbao.com. IN MX 10 mail.matbao.com.  
  Chỉ ra máy chủ mail.matbao.com là 1 Mail Exchanger cho Domain matbao.com với độ ưu tiên là 10.

**7. PTR (Pointer)** : Phân giải địa chỉ IP sang hostname.

* **Cú pháp :**

[Host-ID.{Reverse\_Lookup\_Zone}] IN PTR [tên-máy-tính]

* Ví dụ : record PTR cho host: 112.2.78.100.in-addr.arpa. IN PTR matbao.com.

2. Cài đặt DNS Server

Trong bài này chúng ta cần chuẩn bị 3 máy sau để tiến hành cài đặt DNS Server:

* Primary DNS Server:
* Operating System: CentOS 7 core
* Hostname: primarydns.blogd.local
* IP Address: 10.0.0.1
* Secondary DNS Server:
* Operating System: CentOS 7 cor
* Hostname: secondarydns.blogd.local
* IP Address: 10.0.0.2
* Client:
* Operating System: CentOS 6.5 desktop
* Hostname: clientdns.blogd.local
* IP Address: 10.0.0.3

**Trên Primary DNS Server:**

**Bước 1:** Đầu tiên chúng ta tiến hành cài đặt gói bind bằng cách chạy [lệnh yum](https://blogd.net/linux/cac-thao-tac-co-ban-voi-yum/) như sau:

[root@primarydns ~]# yum -y install bind bind-utils

Loaded plugins: fastestmirror

Loading mirror speeds from cached hostfile

\* base: mirrors.viethosting.com

\* extras: mirrors.viethosting.com

\* updates: mirrors.viethosting.com

Resolving Dependencies

--> Running transaction check

---> Package bind.x86\_64 32:9.9.4-74.el7\_6.2 will be installed

--> Processing Dependency: bind-libs = 32:9.9.4-74.el7\_6.2 for package: 32:bind-9.9.4-74.el7\_6.2.x86\_64

--> Processing Dependency: python-ply for package: 32:bind-9.9.4-74.el7\_6.2.x86\_64

--> Processing Dependency: policycoreutils-python for package: 32:bind-9.9.4-74.el7\_6.2.x86\_64

--> Processing Dependency: policycoreutils-python for package: 32:bind-9.9.4-74.el7\_6.2.x86\_64

--> Processing Dependency: liblwres.so.90()(64bit) for package: 32:bind-9.9.4-74.el7\_6.2.x86\_64

--> Processing Dependency: libisccfg.so.90()(64bit) for package: 32:bind-9.9.4-74.el7\_6.2.x86\_64

--> Processing Dependency: libisccc.so.90()(64bit) for package: 32:bind-9.9.4-74.el7\_6.2.x86\_64

--> Processing Dependency: libisc.so.95()(64bit) for package: 32:bind-9.9.4-74.el7\_6.2.x86\_64

--> Processing Dependency: libdns.so.100()(64bit) for package: 32:bind-9.9.4-74.el7\_6.2.x86\_64

--> Processing Dependency: libbind9.so.90()(64bit) for package: 32:bind-9.9.4-74.el7\_6.2.x86\_64

---> Package bind-utils.x86\_64 32:9.9.4-74.el7\_6.2 will be installed

--> Running transaction check

...

Đặc biệt trong gói bind sẽ chứa file named.root lưu trữ thông tin của **13 root server** nơi chứa các bản ghi gồm toàn bộ các thông tin về tên miền cùng với địa chỉ IP ứng với tên miền đó.

Tiếp theo chúng ta cần thực hiện cấu hình Primary Name Server chúng ta sử dụng [trình soạn thảo văn bản vi](https://blogd.net/linux/su-dung-trinh-soan-thao-vi-vim-co-ban/) để chỉnh sửa file cấu hình chính tên là named.conf (/etc/named.conf):

[root@primarydns ~]# vi /etc/named.conf

//

// named.conf

//

// Provided by Red Hat bind package to configure the ISC BIND named(8) DNS

// server as a caching only nameserver (as a localhost DNS resolver only).

//

// See /usr/share/doc/bind\*/sample/ for example named configuration files.

//

// See the BIND Administrator's Reference Manual (ARM) for details about the

// configuration located in /usr/share/doc/bind-{version}/Bv9ARM.html

options {

listen-on port 53 { 127.0.0.1; };

listen-on-v6 port 53 { ::1; };

directory "/var/named";

dump-file "/var/named/data/cache\_dump.db";

statistics-file "/var/named/data/named\_stats.txt";

memstatistics-file "/var/named/data/named\_mem\_stats.txt";

recursing-file "/var/named/data/named.recursing";

secroots-file "/var/named/data/named.secroots";

allow-query { localhost; };

/\*

- If you are building an AUTHORITATIVE DNS server, do NOT enable recursion.

- If you are building a RECURSIVE (caching) DNS server, you need to enable

recursion.

- If your recursive DNS server has a public IP address, you MUST enable access

control to limit queries to your legitimate users. Failing to do so will

cause your server to become part of large scale DNS amplification

attacks. Implementing BCP38 within your network would greatly

reduce such attack surface

\*/

recursion yes;

dnssec-enable yes;

dnssec-validation yes;

/\* Path to ISC DLV key \*/

bindkeys-file "/etc/named.iscdlv.key";

managed-keys-directory "/var/named/dynamic";

pid-file "/run/named/named.pid";

session-keyfile "/run/named/session.key";

};

logging {

channel default\_debug {

file "data/named.run";

severity dynamic;

};

};

zone "." IN {

type hint;

file "named.ca";

};

include "/etc/named.rfc1912.zones";

include "/etc/named.root.key";

**Bước 2:** Các thông tin cấu hình DNS chúng ta cần thêm các thông tin vào nằm trong phần option:

options {

listen-on port 53 { 127.0.0.1; 10.0.0.1; }; # 10.0.0.1 la dia chi ip cua primary DNS Server

listen-on-v6 port 53 { ::1; };

directory "/var/named";

dump-file "/var/named/data/cache\_dump.db";

statistics-file "/var/named/data/named\_stats.txt";

memstatistics-file "/var/named/data/named\_mem\_stats.txt";

recursing-file "/var/named/data/named.recursing";

secroots-file "/var/named/data/named.secroots";

allow-query { localhost; 10.0.0.0/8; }; # day dia chi nay cho phep truy van DNS server

/\*

- If you are building an AUTHORITATIVE DNS server, do NOT enable recursion.

- If you are building a RECURSIVE (caching) DNS server, you need to enable

recursion.

- If your recursive DNS server has a public IP address, you MUST enable access

control to limit queries to your legitimate users. Failing to do so will

cause your server to become part of large scale DNS amplification

attacks. Implementing BCP38 within your network would greatly

reduce such attack surface

\*/

recursion yes;

dnssec-enable yes; # chuyen tu yes thanh no khi DNS Server fail resolve internet

dnssec-validation yes; # chuyen tu yes thanh no khi DNS Server fail redsolve internet

/\* Path to ISC DLV key \*/

bindkeys-file "/etc/named.iscdlv.key";

managed-keys-directory "/var/named/dynamic";

pid-file "/run/named/named.pid";

session-keyfile "/run/named/session.key";

};

**Lưu ý:** Các file cấu hình có directory là **/var/named**

Trong đó:

* **listen-on port 53**: Đây là địa chỉ ip của máy tính với port đang nghe truy vấn DNS. Chỉ cần thêm địa chỉ ip của máy DNS server là được, còn 127.0.0.1 thì có thể giữ nguyên hoặc bỏ đi.
* **allow-query**: Cho phép những địa chỉ truy vấn. Nếu muốn tất cả đều có thể truy vấn được thì có thể đặt là {any;}.

Khi thông tin mà DNS Server không tìm thấy thì sẽ truy vấn lên 13 root server lớn của thế giới chúng ta có thể thấy zone "." trong file named.conf

zone "." IN {

type hint;

file "named.ca";

};

Thông tin của các root server này được chứa trong file named.ca:

[root@primarydns ~]# cat /var/named/named.ca

; <<>> DiG 9.9.4-RedHat-9.9.4-38.el7\_3.2 <<>> +bufsize=1200 +norec @a.root-servers.net

; (2 servers found)

;; global options: +cmd

;; Got answer:

;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 17380

;; flags: qr aa; QUERY: 1, ANSWER: 13, AUTHORITY: 0, ADDITIONAL: 27

;; OPT PSEUDOSECTION:

; EDNS: version: 0, flags:; udp: 1472

;; QUESTION SECTION:

;. IN NS

;; ANSWER SECTION:

. 518400 IN NS a.root-servers.net.

. 518400 IN NS b.root-servers.net.

. 518400 IN NS c.root-servers.net.

. 518400 IN NS d.root-servers.net.

. 518400 IN NS e.root-servers.net.

. 518400 IN NS f.root-servers.net.

. 518400 IN NS g.root-servers.net.

. 518400 IN NS h.root-servers.net.

. 518400 IN NS i.root-servers.net.

. 518400 IN NS j.root-servers.net.

. 518400 IN NS k.root-servers.net.

. 518400 IN NS l.root-servers.net.

. 518400 IN NS m.root-servers.net.

;; ADDITIONAL SECTION:

a.root-servers.net. 3600000 IN A 198.41.0.4

a.root-servers.net. 3600000 IN AAAA 2001:503:ba3e::2:30

b.root-servers.net. 3600000 IN A 192.228.79.201

b.root-servers.net. 3600000 IN AAAA 2001:500:84::b

c.root-servers.net. 3600000 IN A 192.33.4.12

c.root-servers.net. 3600000 IN AAAA 2001:500:2::c

d.root-servers.net. 3600000 IN A 199.7.91.13

d.root-servers.net. 3600000 IN AAAA 2001:500:2d::d

e.root-servers.net. 3600000 IN A 192.203.230.10

e.root-servers.net. 3600000 IN AAAA 2001:500:a8::e

f.root-servers.net. 3600000 IN A 192.5.5.241

f.root-servers.net. 3600000 IN AAAA 2001:500:2f::f

g.root-servers.net. 3600000 IN A 192.112.36.4

g.root-servers.net. 3600000 IN AAAA 2001:500:12::d0d

h.root-servers.net. 3600000 IN A 198.97.190.53

h.root-servers.net. 3600000 IN AAAA 2001:500:1::53

i.root-servers.net. 3600000 IN A 192.36.148.17

i.root-servers.net. 3600000 IN AAAA 2001:7fe::53

j.root-servers.net. 3600000 IN A 192.58.128.30

j.root-servers.net. 3600000 IN AAAA 2001:503:c27::2:30

k.root-servers.net. 3600000 IN A 193.0.14.129

k.root-servers.net. 3600000 IN AAAA 2001:7fd::1

l.root-servers.net. 3600000 IN A 199.7.83.42

l.root-servers.net. 3600000 IN AAAA 2001:500:9f::42

m.root-servers.net. 3600000 IN A 202.12.27.33

m.root-servers.net. 3600000 IN AAAA 2001:dc3::35

;; Query time: 18 msec

;; SERVER: 198.41.0.4#53(198.41.0.4)

;; WHEN: Po kvě 22 10:14:44 CEST 2017

;; MSG SIZE rcvd: 811

**Bước 3:** Chúng ta cần tạo một zone blogd.server và chứa file phân giải thuận có tên là "blogd.server" dùng để phân giải từ tên miền sang IP:

zone "blogd.server" IN {

type master;

file "blogd.server";

};

Trong đó:

* **zone "blogd.server"**: Là tên miền muốn cấu hình.
* **file "blogd.server"**: Là tên file cấu hình của zone "blogd.net"

Tiếp theo chúng ta cần tạo một zone 0.0.10.in-addr.arpa. Trong zone này sẽ chứa file phân giải nghịch với tên file là "0.0.10.in-addr.arpa" dùng để phân giải từ địa chỉ IP sang tên miền:

zone "0.0.10.in-addr.arpa" IN {

type master;

file "0.0.10.in-addr.arpa";

};

Trong đó:

* **zone "0.0.10.in-addr.arpa"**: Đây chính là tên zone phân giải nghịch. Tên này có quy định là địa chỉ ip của mạng chứa DNS server, nhưng đổi ngược lại. Trong trường hợp này thì địa chỉ mạng là 10.0.0.0 nên khi dùng tạo zone phân giải nghịch ta sẽ lấy ngược lại gồm 3 phần là 0.0.10.
* **file "0.0.10.in-addr.arpa"**: Là tên file cấu hình của zone "0.0.10.in-addr.arpa".

**Bước 4:** Sau đó chúng ta cần phải tạo hai file "blogd.server" va "0.0.10.in-addr.arpa" để có thể thực hiện phân giải tên miền sang ip hay ngược lại:

Chúng ta sử dụng trình soạn thảo vi để tạo vào thêm cấu hình cho file "blogd.server" như sau:

[root@primarydns ~]# vi /var/named/blogd.server

$TTL 86400

@ IN SOA primarydns.blogd.local. root (

2011071001; dns update time for new zone

3600; refresh to new update

1800; retry time for error

604800; expire time after remove record from system

86400; minimun TTL

)

IN NS primarydns.blogd.local.

IN MX 10 primarydns.blogd

1D IN A 10.0.0.1

server1 1D IN A 10.0.0.1

www 1D IN CNAME server1

Sau khi thêm cấu hình file "0.0.10.in-addr.arpa" tiếp theo chúng ta sẽ tạo file và thêm cấu hình vào file "0.0.10.in-addr.arpa" như bên dưới:

[root@primarydns ~]# vi /var/named/0.0.10.in-addr.arpa

$TTL 86400

@ IN SOA primarydns.blogd.local. root. (

2011071001;

3600;

1800;

604800;

86400;

)

@ IN NS primarydns.blogd.local.

1 IN PTR primarydns.blogd.local.

Trong đó:

* **SOA**: Chỉ ra rằng máy chủ Name Server là nơi cung cấp thông tin tin cậy từ dữ liệu có trong zone.
* **NS**: Mỗi name server cho zone sẽ có một NS record.
* **A**: Record A (Address) ánh xạ tên máy (hostname) vào địa chỉ IP.
* **CNAME**: Record CNAME (canonical name) tạo tên bí danh alias trỏ vào một tên canonical.
* **PTR**: Record PTR (pointer) dùng để ánh xạ địa chỉ IP thành hostname.

**Bước 5:** Sau khi thực hiện các bước trên hoàn thành chúng ta sẽ khởi động lại dịch vụ named chạy lệnh bên dưới:

[root@primarydns ~]# systemctl restart named

Sao đó chúng ta cho dịch vụ named khởi động cùng với hệ thống:

[root@primarydns ~]# systemctl enable named

**Bước 6:** Thực hiện truy cập từ client

Trên máy Client chúng ta thực hiện cấu hình lại thông số DNS về 10.0.0.1 như sau:

[root@clientdns ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth1

DEVICE=eth1

HWADDR=00:0C:29:54:B7:2A

TYPE=Ethernet

UUID=c5f55314-5787-459b-83ce-be18b56e1e7e

ONBOOT=yes

NM\_CONTROLLED=yes

BOADCAST=10.255.255.255

IPADDR=10.0.0.3

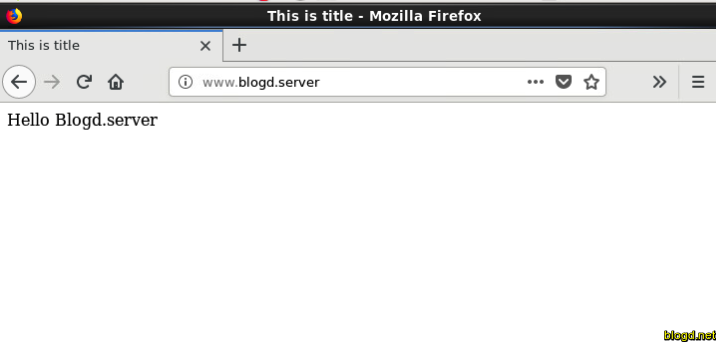
NETMASK=255.0.0.0

NETWORK=10.0.0.0

GATEWAY=10.0.0.1

DNS1=10.0.0.1

Sau đó chúng ta thực hiện mở trình duyệt và truy cập vào tên [http://www.blogd.server:](http://www.blogd.server/)



Trên clientdns chúng ta sử dụng lệnh nslookup để kiểm tra thông tin:

[root@clientdns ~]# nslookup

> www.blogd.server

Server: 10.0.0.1

Address: 10.0.0.1#53

www.blogd.server canonical name = server1.blogd.server.

Name: server1.blogd.server

Address: 10.0.0.1

**Trên Secondary DNS Server:**

**Trên Primary DNS Server:**

**Bước 1:** Cũng tương tự như Primary DNS Server đầu tiên chúng ta tiến hành cài đặt gói bind bằng cách chạy [lệnh yum](https://blogd.net/linux/cac-thao-tac-co-ban-voi-yum/) như sau:

[root@secondarydns ~]# yum -y install bind bind-utils

Loaded plugins: fastestmirror

Loading mirror speeds from cached hostfile

\* base: mirrors.123host.vn

\* extras: mirrors.123host.vn

\* updates: mirrors.123host.vn

base | 3.6 kB 00:00:00

extras | 3.4 kB 00:00:00

updates | 3.4 kB 00:00:00

Resolving Dependencies

--> Running transaction check

---> Package bind.x86\_64 32:9.9.4-74.el7\_6.2 will be installed

--> Processing Dependency: bind-libs = 32:9.9.4-74.el7\_6.2 for package: 32:bind-9.9.4-74.el7\_6.2.x86\_64

--> Processing Dependency: python-ply for package: 32:bind-9.9.4-74.el7\_6.2.x86\_64

--> Processing Dependency: policycoreutils-python for package: 32:bind-9.9.4-74.el7\_6.2.x86\_64

--> Processing Dependency: policycoreutils-python for package: 32:bind-9.9.4-74.el7\_6.2.x86\_64

--> Processing Dependency: liblwres.so.90()(64bit) for package: 32:bind-9.9.4-74.el7\_6.2.x86\_64

--> Processing Dependency: libisccfg.so.90()(64bit) for package: 32:bind-9.9.4-74.el7\_6.2.x86\_64

...

Tiếp theo chúng ta cần thực hiện cấu hình Secondary Name Server chúng ta sử dụng [trình soạn thảo văn bản vi](https://blogd.net/linux/su-dung-trinh-soan-thao-vi-vim-co-ban/) để chỉnh sửa file cấu hình chính tên là named.conf (/etc/named.conf):

//

// See the BIND Administrator's Reference Manual (ARM) for details about the

// configuration located in /usr/share/doc/bind-{version}/Bv9ARM.html

options {

listen-on port 53 { 127.0.0.1; 10.0.0.2; }; # 10.0.0.2 la dia chi cua DNS slave

listen-on-v6 port 53 { ::1; };

directory "/var/named";

dump-file "/var/named/data/cache\_dump.db";

statistics-file "/var/named/data/named\_stats.txt";

memstatistics-file "/var/named/data/named\_mem\_stats.txt";

recursing-file "/var/named/data/named.recursing";

secroots-file "/var/named/data/named.secroots";

allow-query { localhost; 10.0.0.0/8; };

/\*

- If you are building an AUTHORITATIVE DNS server, do NOT enable recursion.

- If you are building a RECURSIVE (caching) DNS server, you need to enable

recursion.

- If your recursive DNS server has a public IP address, you MUST enable access

control to limit queries to your legitimate users. Failing to do so will

cause your server to become part of large scale DNS amplification

attacks. Implementing BCP38 within your network would greatly

reduce such attack surface

\*/

recursion yes;

dnssec-enable yes;

dnssec-validation yes;

/\* Path to ISC DLV key \*/

bindkeys-file "/etc/named.iscdlv.key";

managed-keys-directory "/var/named/dynamic";

pid-file "/run/named/named.pid";

session-keyfile "/run/named/session.key";

};

logging {

channel default\_debug {

file "data/named.run";

severity dynamic;

};

};

zone "." IN {

type hint;

file "named.ca";

};

include "/etc/named.rfc1912.zones";

include "/etc/named.root.key";

Tiếp theo chúng ta cần bổ sung vào file cấu hình Secondary Name Server phần zone "slavethuan" và zone "slavenghich" như bên dưới:

zone "blogd.server" IN {

type slave;

file "slave.thuan";

masters {10.0.0.1;};

};

zone "0.0.10.in-addr.arpa" IN {

type slave;

file "slave.nghich";

masters {10.0.0.1;};

};

**Bước 2:** Sau khi đã chỉnh sửa file cấu hình trên Secondary DNS Server chúng ta tiếp tục quay lại Primary DNS Server bổ sung thông tin allow-transfer vào file /etc/named.conf như bên dưới:

options {

listen-on port 53 { 127.0.0.1; 10.0.0.1; }; # 10.0.0.1 la dia chi ip cua primary DNS Server

listen-on-v6 port 53 { ::1; };

directory "/var/named";

dump-file "/var/named/data/cache\_dump.db";

statistics-file "/var/named/data/named\_stats.txt";

memstatistics-file "/var/named/data/named\_mem\_stats.txt";

recursing-file "/var/named/data/named.recursing";

secroots-file "/var/named/data/named.secroots";

allow-query { localhost; 10.0.0.0/8; }; # day dia chi nay cho phep truy van DNS server

allow-transfer { localhost; 10.0.0.2; }; # dia chi DNS slave

/\*

- If you are building an AUTHORITATIVE DNS server, do NOT enable recursion.

- If you are building a RECURSIVE (caching) DNS server, you need to enable

recursion.

- If your recursive DNS server has a public IP address, you MUST enable access

control to limit queries to your legitimate users. Failing to do so will

cause your server to become part of large scale DNS amplification

attacks. Implementing BCP38 within your network would greatly

reduce such attack surface

\*/

recursion yes;

dnssec-enable yes; # chuyen tu yes thanh no khi DNS Server fail resolve internet

dnssec-validation yes; # chuyen tu yes thanh no khi DNS Server fail redsolve internet

/\* Path to ISC DLV key \*/

bindkeys-file "/etc/named.iscdlv.key";

managed-keys-directory "/var/named/dynamic";

pid-file "/run/named/named.pid";

session-keyfile "/run/named/session.key";

};

**Bước 3:** Khi chúng ta đã chỉnh sửa file cấu hình trên Primary DNS Serve và Secondary DNS Server thì chúng ta thực hiện transfer các tập tin recordbằng cách chạy lệnh sau:

[root@primarydns ~]# rndc reload

server reload successful

[root@secondarydns ~]# rndc reload

server reload successful

Tiếp theo chúng ta khởi động lại named trên cả Primary DNS Serve và Secondary DNS Server:

[root@primarydns ~]# systemctl restart name

[root@secondarydns ~]# systemctl restart named

Thực hiện kiểm tra trên Primary DNS Serve:

[root@primarydns ~]# cd /var/named/

[root@primarydns named]# ls -l

total 24

drwxrwx---. 2 named named 23 Sep 10 15:52 data

drwxrwx---. 2 named named 31 Sep 13 20:55 dynamic

-rw-r-----. 1 root named 2281 May 22 2017 named.ca

-rw-r-----. 1 root named 152 Dec 15 2009 named.empty

-rw-r-----. 1 root named 152 Jun 21 2007 named.localhost

-rw-r-----. 1 root named 168 Dec 15 2009 named.loopback

-rw-r--r--. 1 root root 174 Sep 10 16:27 phangiainghich

-rw-r--r--. 1 root root 412 Sep 10 16:26 phangiaithuan

drwxrwx---. 2 named named 6 Jul 30 00:21 slaves

Và thực hiện trên Secondary DNS Server:

[root@secondarydns ~]# cd /var/named/

[root@secondarydns named]# ls -l

total 24

drwxrwx---. 2 named named 23 Sep 13 20:55 data

drwxrwx---. 2 named named 60 Sep 13 20:56 dynamic

-rw-r-----. 1 root named 2281 May 22 2017 named.ca

-rw-r-----. 1 root named 152 Dec 15 2009 named.empty

-rw-r-----. 1 root named 152 Jun 21 2007 named.localhost

-rw-r-----. 1 root named 168 Dec 15 2009 named.loopback

-rw-r--r--. 1 named named 253 Sep 13 20:55 slave.nghich

drwxrwx---. 2 named named 6 Jul 30 00:21 slaves

-rw-r--r--. 1 named named 527 Sep 13 20:55 slave.thuan

Sau khi kiển tra chúng ta thấy đã có file slaves trên cả 2 DNS Server.

Tiếp theo chúng ta thực hiện sửa thông tin DNS trên clientdns về IP=10.0.0.2.

[root@clientdns ~]# vi /etc/sysconfig/network-scripts/ifcfg-eth1

DEVICE=eth1

HWADDR=00:0C:29:54:B7:2A

TYPE=Ethernet

UUID=c5f55314-5787-459b-83ce-be18b56e1e7e

ONBOOT=yes

NM\_CONTROLLED=yes

BOADCAST=10.255.255.255

IPADDR=10.0.0.3

NETMASK=255.0.0.0

NETWORK=10.0.0.0

GATEWAY=10.0.0.1

DNS1=10.0.0.2

[root@clientdns ~]# systemctl restart network

Mở trình duyệt web vào truy cập thành công web nhu sau:



Trên clientdns chúng ta sử dụng lệnh nslookup để kiểm tra thông tin Slave DNS:

[root@clientdns ~]# nslookup

> www.blogd.server

Server: 10.0.0.2

Address: 10.0.0.2#53

www.blogd.server canonical name = server1.blogd.server.

Name: server1.blogd.server

Address: 10.0.0.1

3. DNS Open-resolver

DNS Open-resolver là một DNS Server sẵn sàng giải quyết các vấn đề tra cứu DNS đệ quy cho bất kỳ ai trên internet. DNS Open-resolver có thể bị lạm dụng cho các cuộc tấn công DDoS.

Để kiểm tra xem DNS server có được định cấu hình cho phép truy vấn đệ quy hay không, chúng ta có thể sử dụng [công cụ dig](https://blogd.net/linux/cac-vi-du-ve-lenh-dig/) để gửi yêu cầu DNS cho tên miền google.com đến địa chỉ IP của DNS Server Primary 10.0.0.1 và đến địa chỉ DNS Server Secondary 10.0.0.2:

[root@primarydns ~]# dig google.com @10.0.0.1

; <<>> DiG 9.11.4-P2-RedHat-9.11.4-9.P2.el7 <<>> google.com @10.0.0.1

;; global options: +cmd

;; Got answer:

;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 19497

;; flags: qr rd ra; QUERY: 1, ANSWER: 6, AUTHORITY: 4, ADDITIONAL: 9

;; OPT PSEUDOSECTION:

; EDNS: version: 0, flags:; udp: 4096

;; QUESTION SECTION:

;google.com. IN A

;; ANSWER SECTION:

google.com. 300 IN A 172.217.194.138

google.com. 300 IN A 172.217.194.101

google.com. 300 IN A 172.217.194.113

google.com. 300 IN A 172.217.194.100

google.com. 300 IN A 172.217.194.102

google.com. 300 IN A 172.217.194.139

;; AUTHORITY SECTION:

google.com. 172797 IN NS ns1.google.com.

google.com. 172797 IN NS ns2.google.com.

google.com. 172797 IN NS ns4.google.com.

google.com. 172797 IN NS ns3.google.com.

;; ADDITIONAL SECTION:

ns2.google.com. 172797 IN A 216.239.34.10

ns1.google.com. 172797 IN A 216.239.32.10

ns3.google.com. 172797 IN A 216.239.36.10

ns4.google.com. 172797 IN A 216.239.38.10

ns2.google.com. 172797 IN AAAA 2001:4860:4802:34::a

ns1.google.com. 172797 IN AAAA 2001:4860:4802:32::a

ns3.google.com. 172797 IN AAAA 2001:4860:4802:36::a

ns4.google.com. 172797 IN AAAA 2001:4860:4802:38::a

;; Query time: 3238 msec

;; SERVER: 10.0.0.1#53(10.0.0.1)

;; WHEN: Tue Sep 24 20:48:55 +07 2019

;; MSG SIZE rcvd: 383

[root@secondarydns ~]# dig google.com @10.0.0.2

; <<>> DiG 9.11.4-P2-RedHat-9.11.4-9.P2.el7 <<>> google.com @10.0.0.2

;; global options: +cmd

;; Got answer:

;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 34890

;; flags: qr rd ra; QUERY: 1, ANSWER: 6, AUTHORITY: 4, ADDITIONAL: 9

;; OPT PSEUDOSECTION:

; EDNS: version: 0, flags:; udp: 4096

;; QUESTION SECTION:

;google.com. IN A

;; ANSWER SECTION:

google.com. 276 IN A 172.217.194.100

google.com. 276 IN A 172.217.194.139

google.com. 276 IN A 172.217.194.102

google.com. 276 IN A 172.217.194.113

google.com. 276 IN A 172.217.194.138

google.com. 276 IN A 172.217.194.101

;; AUTHORITY SECTION:

google.com. 172775 IN NS ns4.google.com.

google.com. 172775 IN NS ns3.google.com.

google.com. 172775 IN NS ns2.google.com.

google.com. 172775 IN NS ns1.google.com.

;; ADDITIONAL SECTION:

ns2.google.com. 172775 IN A 216.239.34.10

ns1.google.com. 172775 IN A 216.239.32.10

ns3.google.com. 172775 IN A 216.239.36.10

ns4.google.com. 172775 IN A 216.239.38.10

ns2.google.com. 172775 IN AAAA 2001:4860:4802:34::a

ns1.google.com. 172775 IN AAAA 2001:4860:4802:32::a

ns3.google.com. 172775 IN AAAA 2001:4860:4802:36::a

ns4.google.com. 172775 IN AAAA 2001:4860:4802:38::a

;; Query time: 1 msec

;; SERVER: 10.0.0.2#53(10.0.0.2)

;; WHEN: Tue Sep 24 20:56:14 +07 2019

;; MSG SIZE rcvd: 383

Qua lệnh trên chúng ta thu được kết quả của DNS Server Primary:

;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 19497

;; flags: qr rd ra; QUERY: 1, ANSWER: 6, AUTHORITY: 4, ADDITIONAL: 9

Và kết quả của DNS Server Secondary:

;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 34890

;; flags: qr rd ra; QUERY: 1, ANSWER: 6, AUTHORITY: 4, ADDITIONAL: 9

Kết quả trên cho chúng ta biết DNS server trên là một open resolver mở cho phép truy vấn đệ quy.

Nếu khi chúng ta thực hiện kiểm tra và nhận được kết quả:

[root@primarydns ~]# dig google.com @10.0.0.1

; <<>> DiG 9.11.4-P2-RedHat-9.11.4-9.P2.el7 <<>> google.com @10.0.0.1

;; global options: +cmd

;; Got answer:

;; ->>HEADER<<- opcode: QUERY, status: REFUSED, id: 3245

;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:

; EDNS: version: 0, flags:; udp: 4096

;; QUESTION SECTION:

;google.com. IN A

;; Query time: 1 msec

;; SERVER: 10.0.0.1#53(10.0.0.1)

;; WHEN: Tue Sep 24 23:38:50 +07 2019

;; MSG SIZE rcvd: 39

Kết quả này cho chúng ta biết DNS Server không cho phép truy vấn đệ quy.

Để có thể bật tắt truy vấn đệ quy chúng ta cần chỉnh sửa trong file cấu hình named.conf:

* Bật truy vấn đệ quy chúng ta chỉnh như sau:
* recursion yes;
* Tắt truy vấn đệ quy chúng ta chỉnh như sau:
* recursion no;

**Lưu ý:** Khi chúng ta bật truy vấn đệ quy thì cần bổ sung cấu hình rate limit để hạn chế bị lợi dụng tấn công DNS Amplification như sau:

rate-limit {

responses-per-second 10;

log-only yes;

}

4. Giới thiệu về lệnh dig

Domain Information Groper(dig) là một công cụ mạng mạnh mẽ để truy vấn máy chủ tên DNS.

Lệnh dig có thể truy vấn thông tin về các bản ghi DNS khác nhau bao gồm địa chỉ máy chủ. Đây là công cụ được sử dụng phổ biến để khắc phục sự cố DNS vì tính linh hoạt và dễ sử dụng của nó.

5. Cài đặt dig – Công cụ

Để kiểm tra xem lệnh dig có khả dụng trên loại hệ thống của bạn không sử dụng lệnh sau:

dig -v

Nếu lệnh dig không có trên hệ thống của bạn, bạn có thể dễ dàng cài đặt lệnh dig như sau:

* Đối với bản phân phối Debian/Ubuntu:
* apt install dnsutils
* Đối với với bản phân phối RHEL/CentOS:
* yum install bind-utils

6. Cách sử dụng dig

**6.1. Truy vấn DNS cơ bản**

Trong ví dụ sau, chúng ta sẽ thực hiện một truy vấn để lấy thông tin tên miền google.com. Chúng ta thực hiện như sau:

[root@test1 ~]# dig google.com

; <<>> DiG 9.9.4-RedHat-9.9.4-73.el7\_6 <<>> google.com

;; global options: +cmd

;; Got answer:

;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 21853

;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:

; EDNS: version: 0, flags:; udp: 1452

;; QUESTION SECTION:

;google.com. IN A

;; ANSWER SECTION:

google.com. 271 IN A 216.58.220.206

;; Query time: 36 msec

;; SERVER: 1.1.1.1#53(1.1.1.1)

;; WHEN: Mon Apr 15 15:01:53 +07 2019

;; MSG SIZE rcvd: 55

Ý nghĩa các phần:

* Dòng đầu tiên của đầu ra hiển thị version đã cài đặt và truy vấn được gọi. Dòng thứ hai hiển thị các tùy chọn(theo mặc định chỉ cmd).
* ; <<>> DiG 9.9.4-RedHat-9.9.4-73.el7\_6 <<>> google.com
* ;; global options: +cmd
* Phần tiếp thep, dòng đầu tiên của phần này là tiêu đề, bao gồm opcode và trạng thái của hành động. Trong trường hợp này, trạng thái NOERROR có nghĩa là yêu cầu truy vấn truy vấn DNS không gặp lỗi.
* ;; Got answer:
* ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 21853
* ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
* Phần này được hiển thị theo mặc định chỉ trên các phiên bản mới hơn.
* ;; OPT PSEUDOSECTION:
* ; EDNS: version: 0, flags:; udp: 1452
* Đây là phần mà lệnh dig hiển thị truy vấn của chúng ta. Theo mặc định, dig sẽ yêu cầu bản ghi A.
* ;; QUESTION SECTION:
* ;google.com. IN A
* Phần trả lời cung cấp cho chúng ta tên miền google.com. trỏ đến địa chỉ IP 216.58.220.206.
* ;; ANSWER SECTION:
* google.com. 271 IN A 216.58.220.206
* Đây là phần cuối cùng của đầu ra bao gồm số liệu thống kê về truy vấn.
* ;; Query time: 36 msec
* ;; SERVER: 1.1.1.1#53(1.1.1.1)
* ;; WHEN: Mon Apr 15 15:01:53 +07 2019
* ;; MSG SIZE rcvd: 55

Trong đó :

Query time: Cho biết thời gian của kết nối tốn bao lâu.

SERVER: 1.1.1.1: Cho biết name resolver đang dùng.

WHEN: Thời gian thực hiện truy vấn DNS.

MSG SIZE rcvd: Kích thước gói tin trả lời truy vấn DNS.

**6.2. Truy vấn địa chỉ ip**

Để có kết quả ngắn cho truy vấn của bạn, hãy sử dụng tùy chọn +short tùy chọn:

[root@test1 ~]# dig amazon.com +short

205.251.242.103

176.32.98.166

176.32.103.205

Kết quả chỉ bao gồm các địa chỉ IP của bản ghi A.

Để có kết quả chi tiết hơn bằng cách sử dụng các tùy chọn +noall và tùy chọn +answer.

[root@test1 ~]# dig amazon.com +noall +answer

; <<>> DiG 9.9.4-RedHat-9.9.4-73.el7\_6 <<>> amazon.com +noall +answer

;; global options: +cmd

amazon.com. 25 IN A 205.251.242.103

amazon.com. 25 IN A 176.32.98.166

amazon.com. 25 IN A 176.32.103.205

Trong đó:

* +noall: Tắt tất cả các section
* +answer: Chỉ mở answer section

**6.3. Truy vấn bản ghi MX cho tên miền**

Chỉ truy vấn các loại bản ghi MX cho tên miền. Chúng ta thực hiện như sau:

[root@test1 ~]# dig amazon.com MX +noall +answer

; <<>> DiG 9.9.4-RedHat-9.9.4-73.el7\_6 <<>> amazon.com MX +noall +answer

;; global options: +cmd

amazon.com. 900 IN MX 5 amazon-smtp.amazon.com.

**6.4. Truy vấn bản ghi SOA cho tên miền**

Chỉ truy vấn các loại bản ghi SOA cho tên miền. Chúng ta thực hiện như sau:

[root@test1 ~]# dig amazon.com SOA +noall +answer

; <<>> DiG 9.9.4-RedHat-9.9.4-73.el7\_6 <<>> amazon.com SOA +noall +answer

;; global options: +cmd

amazon.com. 39 IN SOA dns-external-master.amazon.com. root.amazon.com. 2010115712 180 60 3024000 60

**6.5. Truy vấn Bản ghi TTL cho tên miền**

Chỉ truy vấn các loại bản ghi TTL cho tên miền. Chúng ta thực hiện như sau:

[root@test1 ~]# dig amazon.com TTL +noall +answer

; <<>> DiG 9.9.4-RedHat-9.9.4-73.el7\_6 <<>> amazon.com TTL +noall +answer

;; global options: +cmd

;; Got answer:

;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 8008

;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:

; EDNS: version: 0, flags:; udp: 1452

;; QUESTION SECTION:

;amazon.com. IN A

;; ANSWER SECTION:

amazon.com. 18 IN A 176.32.103.205

amazon.com. 18 IN A 205.251.242.103

amazon.com. 18 IN A 176.32.98.166

;; Query time: 34 msec

;; SERVER: 1.1.1.1#53(1.1.1.1)

;; WHEN: Mon Apr 15 23:21:02 +07 2019

;; MSG SIZE rcvd: 87

**6.6. Tra cứu DNS ngược**

Để truy vấn tên máy chủ được liên kết với một địa chỉ IP cụ thể, hãy sử dụng tùy chọn -x.

Ví dụ: Để thực hiện tra cứu ngược lại địa chỉ ip 31.13.95.36 thành tên miền chúng ta thực hiện:

[root@test1 ~]# dig -x 31.13.95.36 +noall +answer

; <<>> DiG 9.9.4-RedHat-9.9.4-73.el7\_6 <<>> -x 31.13.95.36 +noall +answer

;; global options: +cmd

36.95.13.31.in-addr.arpa. 2863 IN PTR edge-star-mini-shv-01-hkg3.facebook.com.

**6.7. Truy vấn số lượng lớn tên miền**

Nếu bạn muốn truy vấn một số lượng lớn tên miền, bạn có thể thêm chúng vào một tệp và sử dụng tùy chọn -f để truy vấn.

Trong ví dụ sau, chúng tôi đang truy vấn các miền được liệt kê trong tệp tenmien.txt.

[root@test1 ~]# cat tenmien.txt

google.com

gmail.com

linux.org

[root@test1 ~]# dig -f tenmien.txt +noall +answer

google.com. 123 IN A 216.58.200.14

gmail.com. 131 IN A 172.217.31.229

linux.org. 300 IN A 104.27.166.219

linux.org. 300 IN A 104.27.167.219

**6.8. Truy vấn DNS bằng TCP**

Theo mặc định thì lệnh dig truy vấn DNS qua udp, để có thể dùng lệnh dig truy vấn DNS qua tcp chúng ta thực hiện như sau:

dig +tcp [domain]

**6.9. Điều chỉnh mặc định với tệp ~/.digrc**

Chúng ta có thể tạo tệp .digrc trong thư mục chính để bao gồm tùy chọn tùy mà chúng ta muốn lệnh dig thực hiện. Chúng ta có thể chỉ định các tùy chọn khác nhau trong tệp ~/.digrc sẽ luôn tự động chạy khi sử dụng lệnh dig.

Trong ví dụ dưới đây, chúng ta thêm tùy chọn +short vào ~/.digrc sau đó thực hiện lệnh dig cuúng ta thực hiện như sau:

[root@test1 ~]# cat .digrc

+short

[root@test1 ~]# dig amazon.com

176.32.98.166

176.32.103.205

205.251.242.103