How to Setup DNS Server (Bind) on CentOS 8 / RHEL8

Pradeep Kumar / October 25, 2020

Developed in the 80's by students at Berkeley University, **BIND** (**Berkeley Internet Name Domain**) is an open source **DNS server** that provides DNS services on Linux distributions. So, what is a DNS server ? A **DNS** server is a service that helps to resolve a fully qualified domain name (**FQDN**) into an IP address and additionally, perform a reverse translation-translation of an IP address to a user-friendly domain name.

Why is name resolution important? Well, computers locate services on servers using IP addresses. However, IP addresses are not as user-friendly as domain names and it would be a big headache trying to remember each IP address that is associated with every domain name. A DNS server steps in and helps to resolve these domain names to computer IP addresses.

This guide walks you through the process of setting up a DNS bind server on CentOS 8 / RHEL 8.

Lab setup:

- Server : CentOS 8 (minimal server)
- IP address : 192.168.43.35
- Hots Name : dns-primary.linuxtechi.local
- Domain : linuxtechi.local

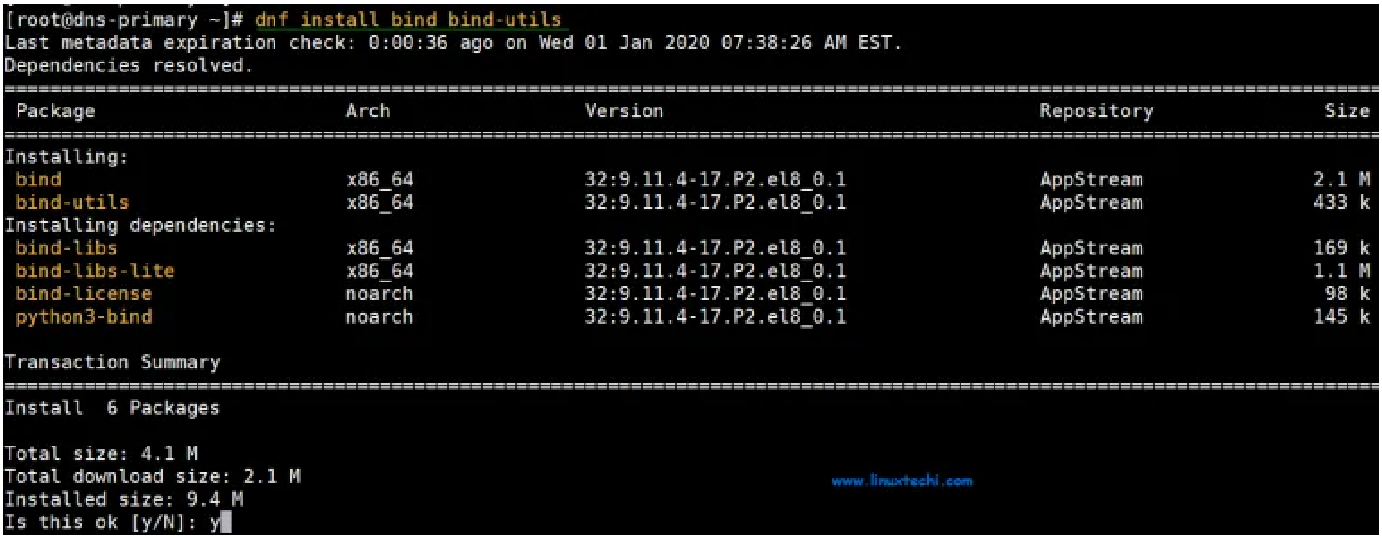
Let's now hit the ground running with the configuration of the DNS bind server.

Step 1: Install bind DNS on CentOS 8 / RHEL 8

We begin with the installation of the bind and bind-utils package. These packages constitutes dns server and its utilities responsible for querying name servers or DNS servers.

Execute the command:

```
# dnf install bind bind-utils
```



```
[root@dns-primary ~]# dnf install bind bind-utils
Last metadata expiration check: 0:00:36 ago on Wed 01 Jan 2020 07:38:26 AM EST.
Dependencies resolved.
================================================================================================================
 Package               Arch            Version                         Repository              Size
================================================================================================================
Installing:
 bind                  x86_64          32:9.11.4-17.P2.el8_0.1         AppStream               2.1 M
 bind-utils            x86_64          32:9.11.4-17.P2.el8_0.1         AppStream               433 k
Installing dependencies:
 bind-libs             x86_64          32:9.11.4-17.P2.el8_0.1         AppStream               169 k
 bind-libs-lite        x86_64          32:9.11.4-17.P2.el8_0.1         AppStream               1.1 M
 bind-license          noarch          32:9.11.4-17.P2.el8_0.1         AppStream                98 k
 python3-bind          noarch          32:9.11.4-17.P2.el8_0.1         AppStream               145 k

Transaction Summary
================================================================================================================
Install  6 Packages

Total size: 4.1 M
Total download size: 2.1 M
Installed size: 9.4 M
Is this ok [y/N]: y
```

Once successfully installed, start the DNS server using the command below:

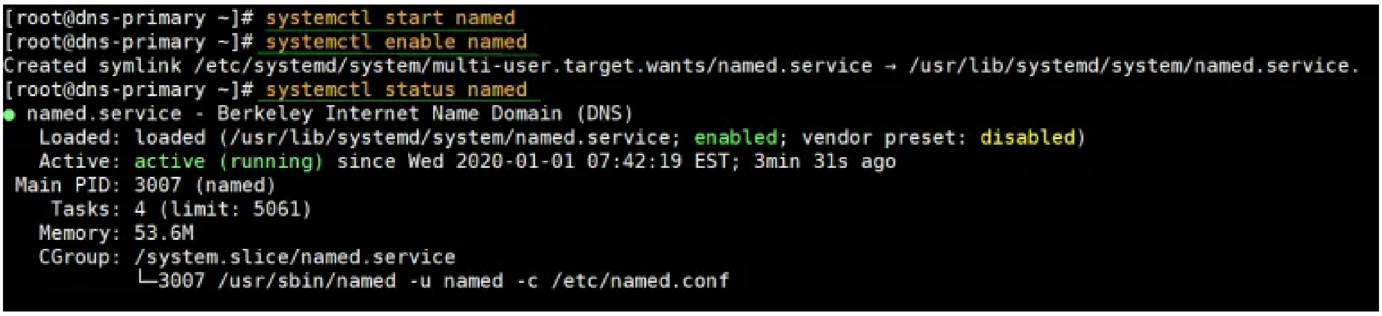
```
# systemctl start named
```

Next, enable it so that it can kick in even after a reboot

```
# systemctl enable named
```

Just to be sure that the service is running as expected, check its status

```
# systemctl status named
```



```
[root@dns-primary ~]# systemctl start named
[root@dns-primary ~]# systemctl enable named
Created symlink /etc/systemd/system/multi-user.target.wants/named.service → /usr/lib/systemd/system/named.service.
[root@dns-primary ~]# systemctl status named
● named.service - Berkeley Internet Name Domain (DNS)
   Loaded: loaded (/usr/lib/systemd/system/named.service; enabled; vendor preset: disabled)
   Active: active (running) since Wed 2020-01-01 07:42:19 EST; 3min 31s ago
 Main PID: 3007 (named)
    Tasks: 4 (limit: 5061)
   Memory: 53.6M
   CGroup: /system.slice/named.service
           └─3007 /usr/sbin/named -u named -c /etc/named.conf
```

Great, the DNS server is running just perfectly. Now let's jump into configuring the Bind DNS server

Step 2: Configure bind DNS server

Usually, best practice recommends making a backup of a configuration file before making any changes. This is so that should anything go wrong, we can always revert to the original unedited file. And it’s no different here.

Let’s take a backup of the config file /etc/named.conf

```
# cp /etc/named.conf  /etc/named.bak
```

Now go ahead and open the file using your preferred text editor. In this case, we’re using vim editor.

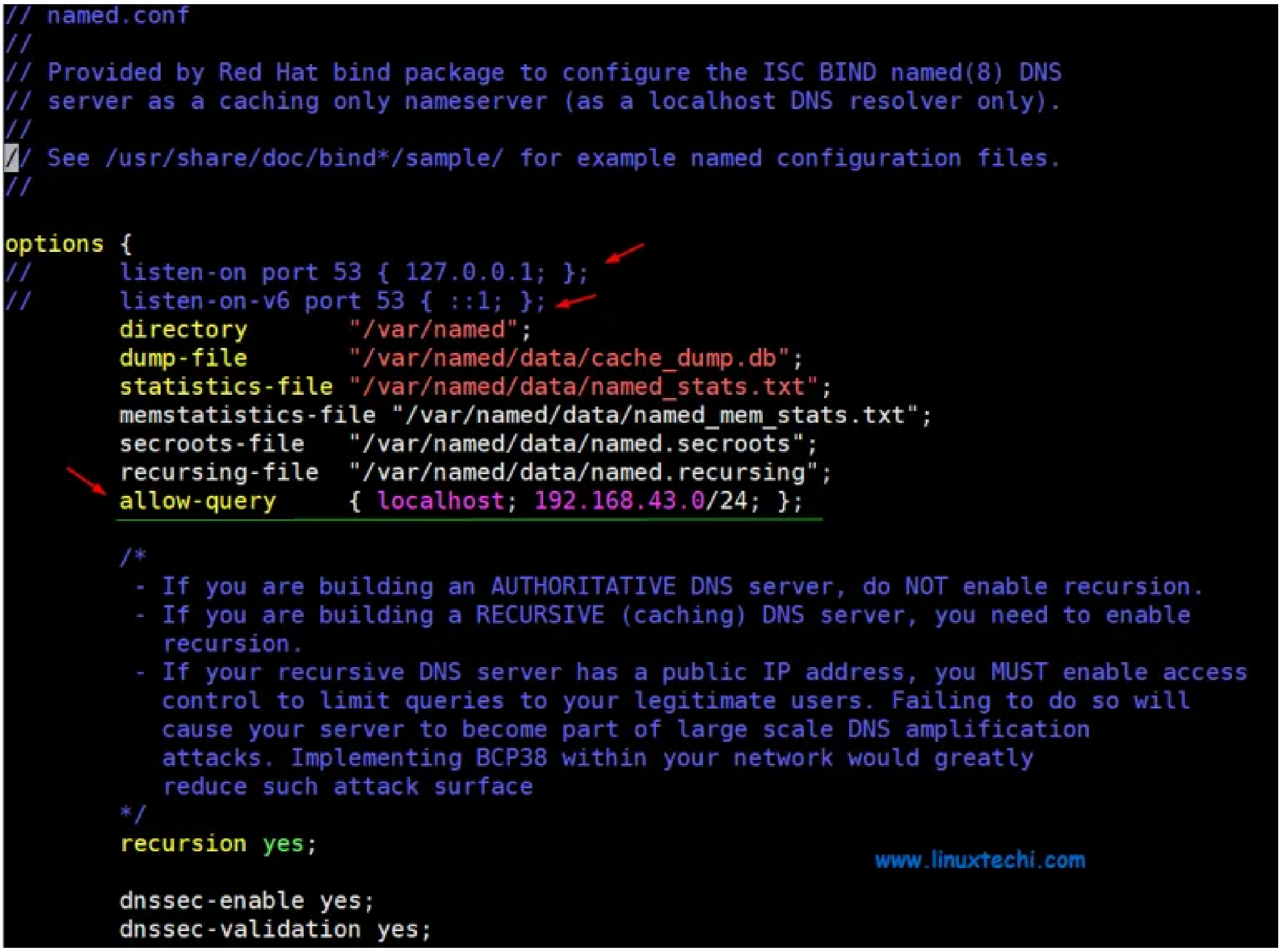
```
# vim /etc/named.conf
```

Under the ‘**Options**’ section, ensure you comment out the lines indicated below to enable the Bind DNS server to listen to all IPs.

```
// listen-on port 53 { 127.0.0.1; };
// listen-on-v6 port 53 { ::1; };
```

Additionally, locate the allow-query parameter and adjust it according to your network subnet.

```
allow-query { localhost; 192.168.43.0/24; };
```



```
// 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.
//

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";
        secroots-file   "/var/named/data/named.secroots";
        recursing-file  "/var/named/data/named.recursing";
        allow-query     { localhost; 192.168.43.0/24; };

        /*
         - 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;
```

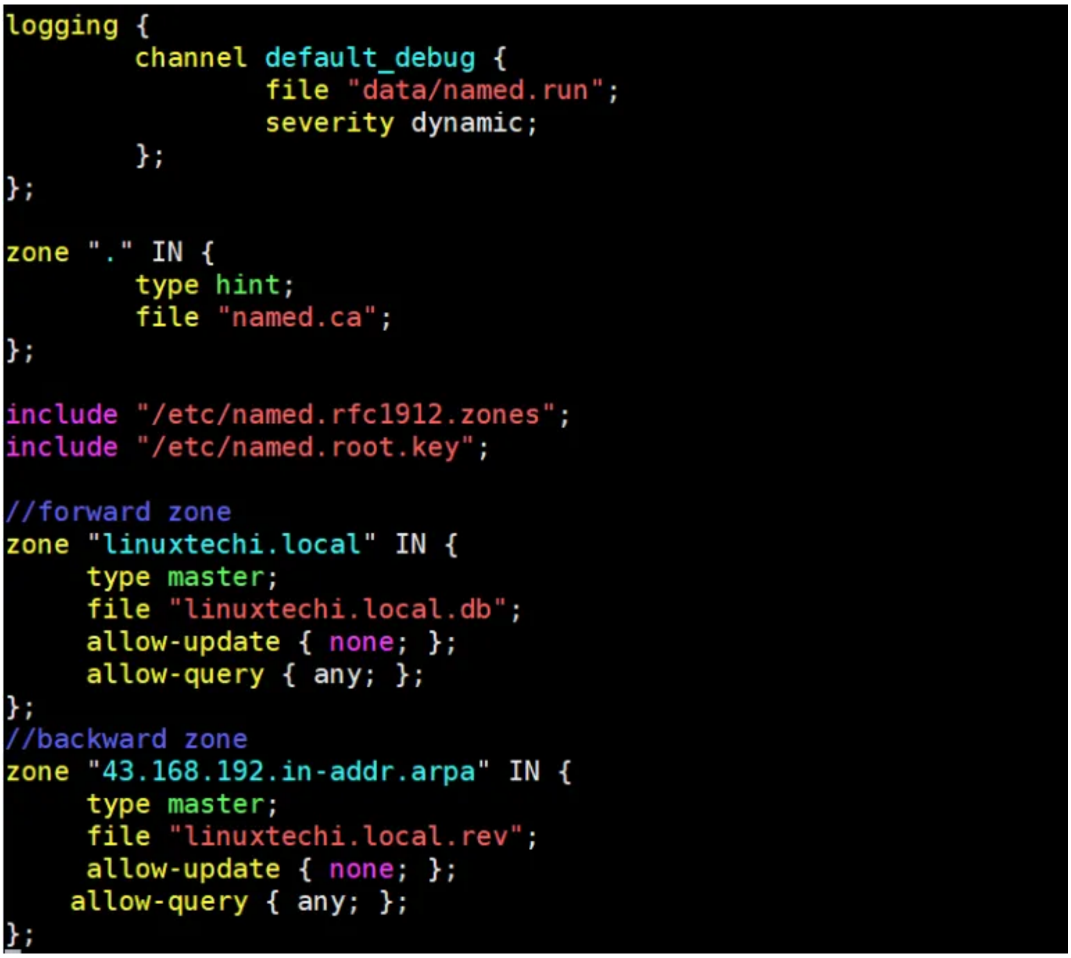
This setting allows only the hosts in the defined network to access the DNS server and not just any other host.

A **forward lookup DNS zone** is one that stores the host name ip address relationship. When queried, it gives the IP address of the host system using the host name. In contrast, the reverse **DNS zone** returns the Fully Qualified Domain Name (FQDN) of the server in relation to it’s IP address.

To define the reverse and forward lookup zones, copy and paste the following configuration at the end of **/etc/named.conf**

```
//forward zone

zone "linuxtechi.local" IN {

     type master;

     file "linuxtechi.local.db";

     allow-update { none; };

     allow-query { any; };

};
```

```
//backward zone
zone "43.168.192.in-addr.arpa" IN {
     type master;
     file "linuxtechi.local.rev";
     allow-update { none; };
     allow-query { any; };
};
```



```
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";

//forward zone
zone "linuxtechi.local" IN {
     type master;
     file "linuxtechi.local.db";
     allow-update { none; };
     allow-query { any; };
};
//backward zone
zone "43.168.192.in-addr.arpa" IN {
     type master;
     file "linuxtechi.local.rev";
     allow-update { none; };
    allow-query { any; };
};
```

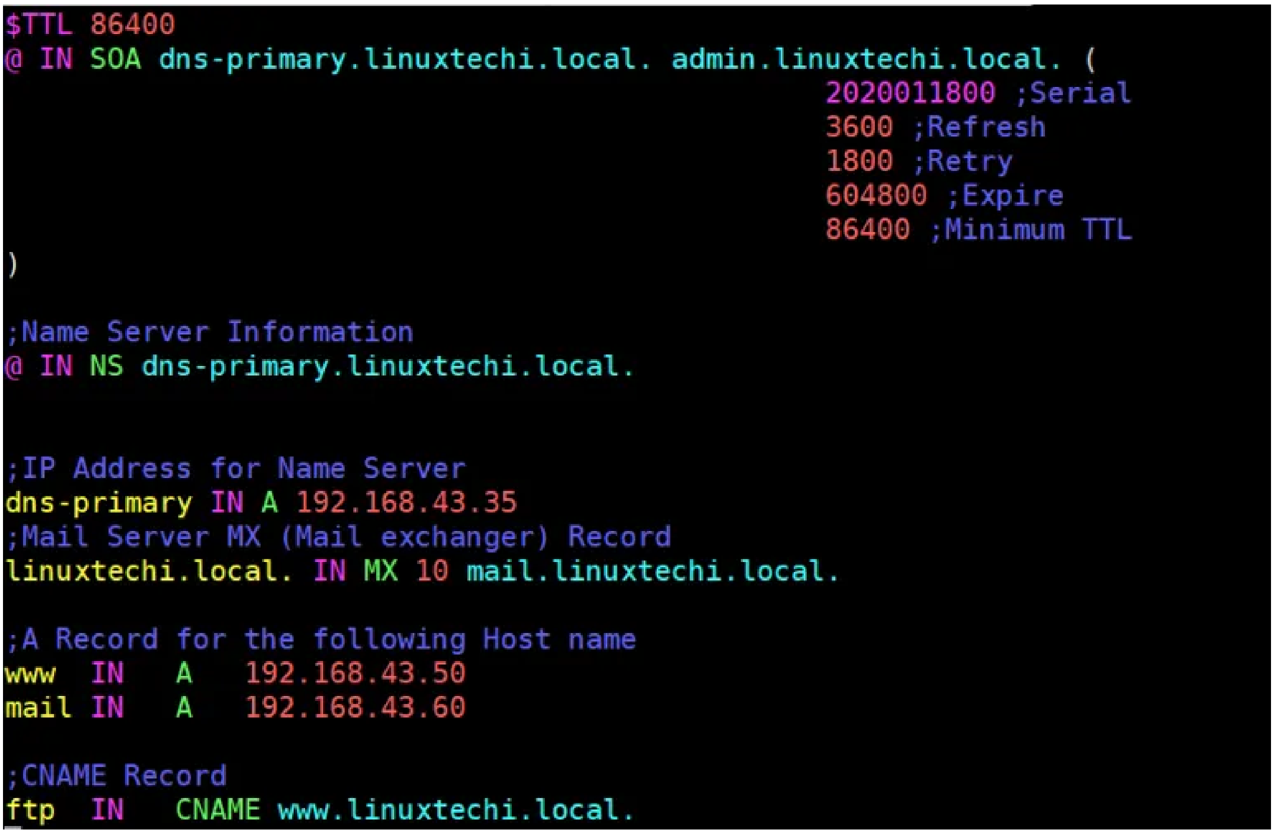
- **type**: Stipulates the role of the server for a particular zone. the attribute 'master' implies that this is an authoritative server.
- **file**: Points to the forward / reverse zone file of the domain.
- **allow-update**: This attribute defined the host systems which are permitted to forward Dynamic DNS updates. In this case, we don't have any.

After saving, exit the /etc/named.conf configuration file.

Step 3: Create a forward DNS zone file for the domain

Create a forward DNS zone file for domain linuxtechi.local as shown below and add the following content

```
[root@dns-primary ~]# vim /var/named/linuxtechi.local.db
$TTL 86400
@ IN SOA dns-primary.linuxtechi.local. admin.linuxtechi.local. (
                                                2020011800 ;Serial
                                                3600 ;Refresh
                                                1800 ;Retry
                                                604800 ;Expire
                                                86400 ;Minimum TTL
)

;Name Server Information
@ IN NS dns-primary.linuxtechi.local.

;IP Address for Name Server
dns-primary IN A 192.168.43.35

;Mail Server MX (Mail exchanger) Record
linuxtechi.local. IN MX 10 mail.linuxtechi.local.

;A Record for the following Host name
www  IN   A   192.168.43.50
mail IN   A   192.168.43.60

;CNAME Record
ftp  IN   CNAME www.linuxtechi.local.
```



```
$TTL 86400
@ IN SOA dns-primary.linuxtechi.local. admin.linuxtechi.local. (
                                                2020011800 ;Serial
                                                3600 ;Refresh
                                                1800 ;Retry
                                                604800 ;Expire
                                                86400 ;Minimum TTL
)

;Name Server Information
@ IN NS dns-primary.linuxtechi.local.


;IP Address for Name Server
dns-primary IN A 192.168.43.35
;Mail Server MX (Mail exchanger) Record
linuxtechi.local. IN MX 10 mail.linuxtechi.local.

;A Record for the following Host name
www  IN   A   192.168.43.50
mail IN   A   192.168.43.60

;CNAME Record
ftp  IN   CNAME www.linuxtechi.local.
```

Let's define some of the parameters indicated in the configuration file:

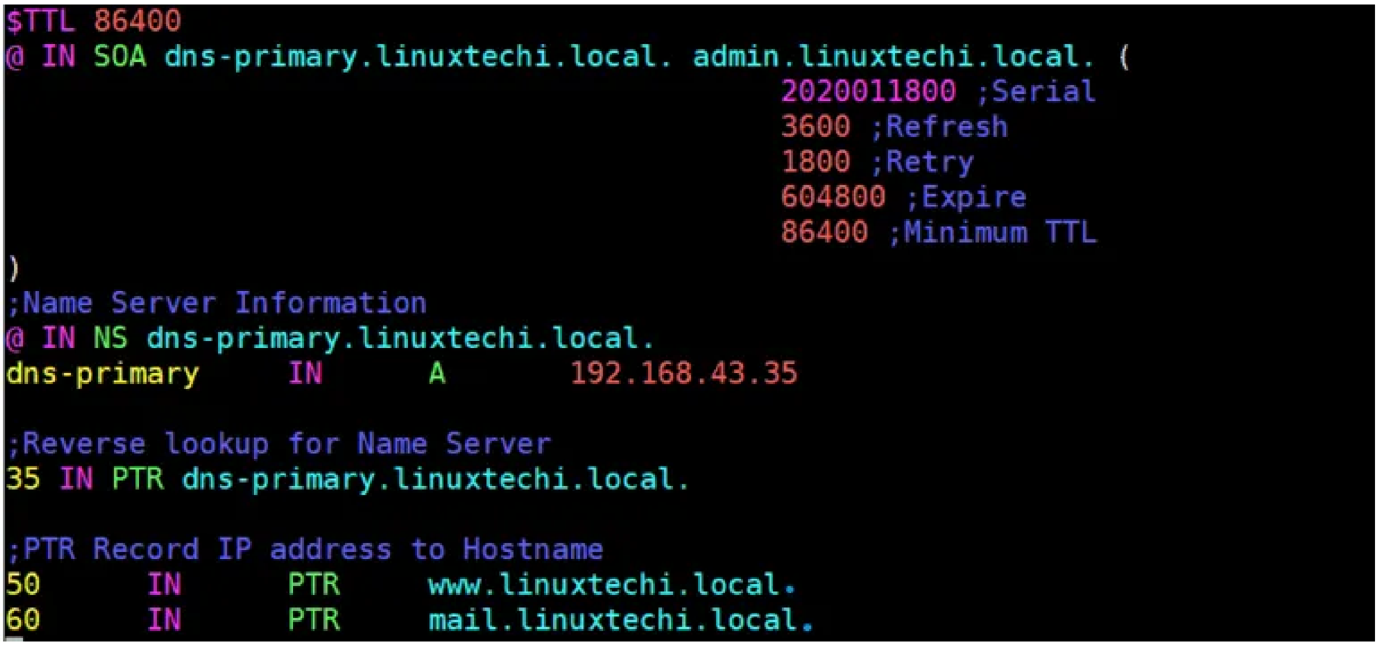
- **TTL**: This is short for Time-To-Live. TTL is the duration of time (or hops) that a packet exists in a network before finally being discarded by the router.
- **IN**: This implies the Internet.
- **SOA**: This is short for the Start of Authority. Basically, it defines the authoritative name server, in this case, dns-primary.linuxtechi.local and contact information – admin.linuxtechi.local
- **NS**: This is short for Name Server.
- **A**: This is an A record. It points to a domain/subdomain name to the IP Address
- **Serial**: This is the attribute used by the DNS server to ensure that contents of a specific zone file are updated.
- **Refresh**: Defines the number of times that a slave DNS server should transfer a zone from the master.
- **Retry**: Defines the number of times that a slave should retry a non-responsive zone transfer.
- **Expire**: Specifies the duration a slave server should wait before responding to a client query when the Master is unavailable.
- **Minimum**: This is responsible for setting the minimum TTL for a zone.
- **MX**: This is the Mail exchanger record. It specifies the mail server receiving and sending emails

- **CNAME**: This is the Canonical Name. It maps an alias domain name to another domain name.
- **PTR**: Short for Pointer, this attributes resolves an IP address to a domain name, opposite to a domain name.

Step 4: Create a reverse DNS zone file for the domain

Similarly, we need to create a file for reverse DNS lookups. Paste the following content in the file:

```
[root@dns-primary ~]# vim /var/named/linuxtechi.local.rev
$TTL 86400
@ IN SOA dns-primary.linuxtechi.local. admin.linuxtechi.local. (
                                            2020011800 ;Serial
                                            3600 ;Refresh
                                            1800 ;Retry
                                            604800 ;Expire
                                            86400 ;Minimum TTL
)
;Name Server Information
@ IN NS dns-primary.linuxtechi.local.
dns-primary     IN      A       192.168.43.35

;Reverse lookup for Name Server
35 IN PTR dns-primary.linuxtechi.local.

;PTR Record IP address to Hostname
50      IN      PTR     www.linuxtechi.local.
60      IN      PTR     mail.linuxtechi.local.
```



```
$TTL 86400
@ IN SOA dns-primary.linuxtechi.local. admin.linuxtechi.local. (
                                            2020011800 ;Serial
                                            3600 ;Refresh
                                            1800 ;Retry
                                            604800 ;Expire
                                            86400 ;Minimum TTL
)
;Name Server Information
@ IN NS dns-primary.linuxtechi.local.
dns-primary     IN      A       192.168.43.35

;Reverse lookup for Name Server
35 IN PTR dns-primary.linuxtechi.local.

;PTR Record IP address to Hostname
50      IN      PTR     www.linuxtechi.local.
60      IN      PTR     mail.linuxtechi.local.
```

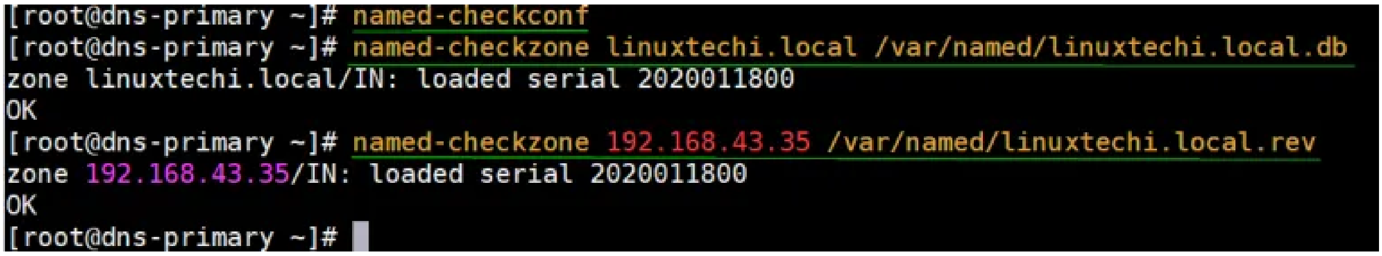
Next, assign the necessary file permissions to the two configuration files.

```
# chown named:named /var/named/linuxtechi.local.db

# chown named:named /var/named/linuxtechi.local.rev
```

To confirm that the DNS zone lookup files are free from any syntactical errors, run the commands shown:

```
# named-checkconf

# named-checkzone linuxtechi.local /var/named/linuxtechi.local.db

# named-checkzone 192.168.43.35 /var/named/linuxtechi.local.rev
```

If there are no errors, you should get the output shown:



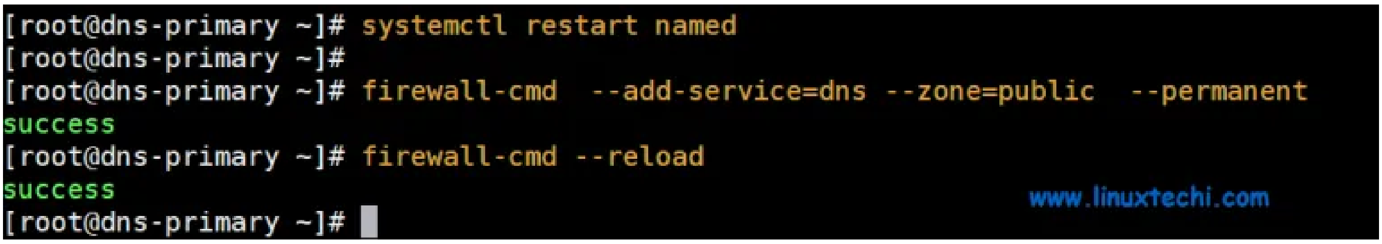
```
[root@dns-primary ~]# named-checkconf
[root@dns-primary ~]# named-checkzone linuxtechi.local /var/named/linuxtechi.local.db
zone linuxtechi.local/IN: loaded serial 2020011800
OK
[root@dns-primary ~]# named-checkzone 192.168.43.35 /var/named/linuxtechi.local.rev
zone 192.168.43.35/IN: loaded serial 2020011800
OK
[root@dns-primary ~]#
```

For the changes to be reflected in the system, restart the Bind DNS server

```
# systemctl restart named
```

For client systems to access the system we need to add the DNS service on the firewall and thereafter reload the firewall.

```
# firewall-cmd  --add-service=dns --zone=public  --permanent

# firewall-cmd --reload
```



```
[root@dns-primary ~]# systemctl restart named
[root@dns-primary ~]#
[root@dns-primary ~]# firewall-cmd  --add-service=dns --zone=public  --permanent
success
[root@dns-primary ~]# firewall-cmd --reload
success
[root@dns-primary ~]# 
```

www.linuxtechi.com

Step 5: Test the Bind DNS server from a client system

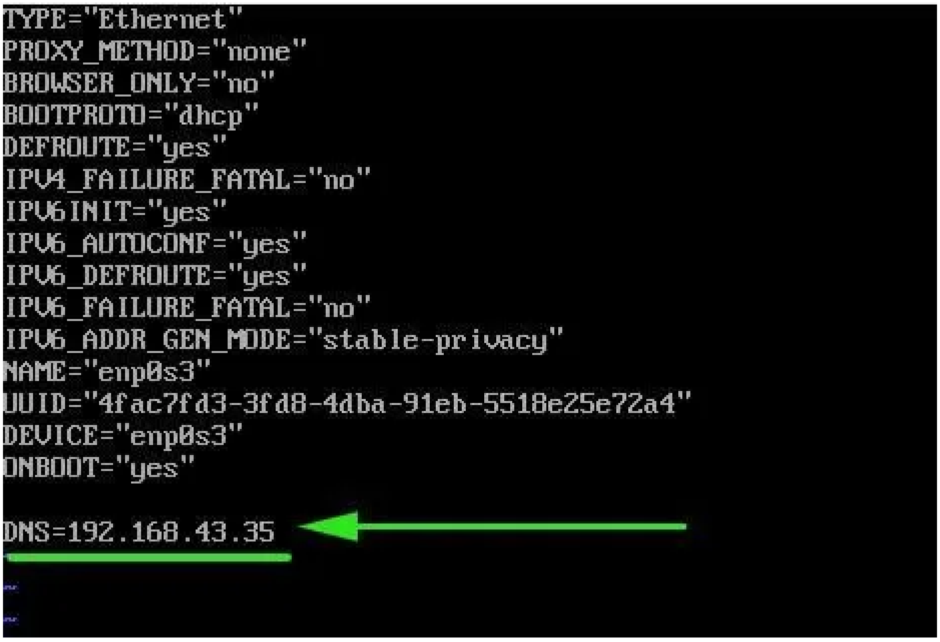
Having finalized the configuration on the DNS server, let’s head out to a client machine and perform a few tests.

On a client machine (CentOS 8 / RHEL 8), open the **/etc/resolv.conf** file and edit the following parameter:

nameserver 192.168.43.35

As always, save & close the configuration file.

Lastly, you need to append the Bind DNS server’s IP address to the **/etc/sysconfig/network-scripts/ifcfg-enp0s3** file as shown.



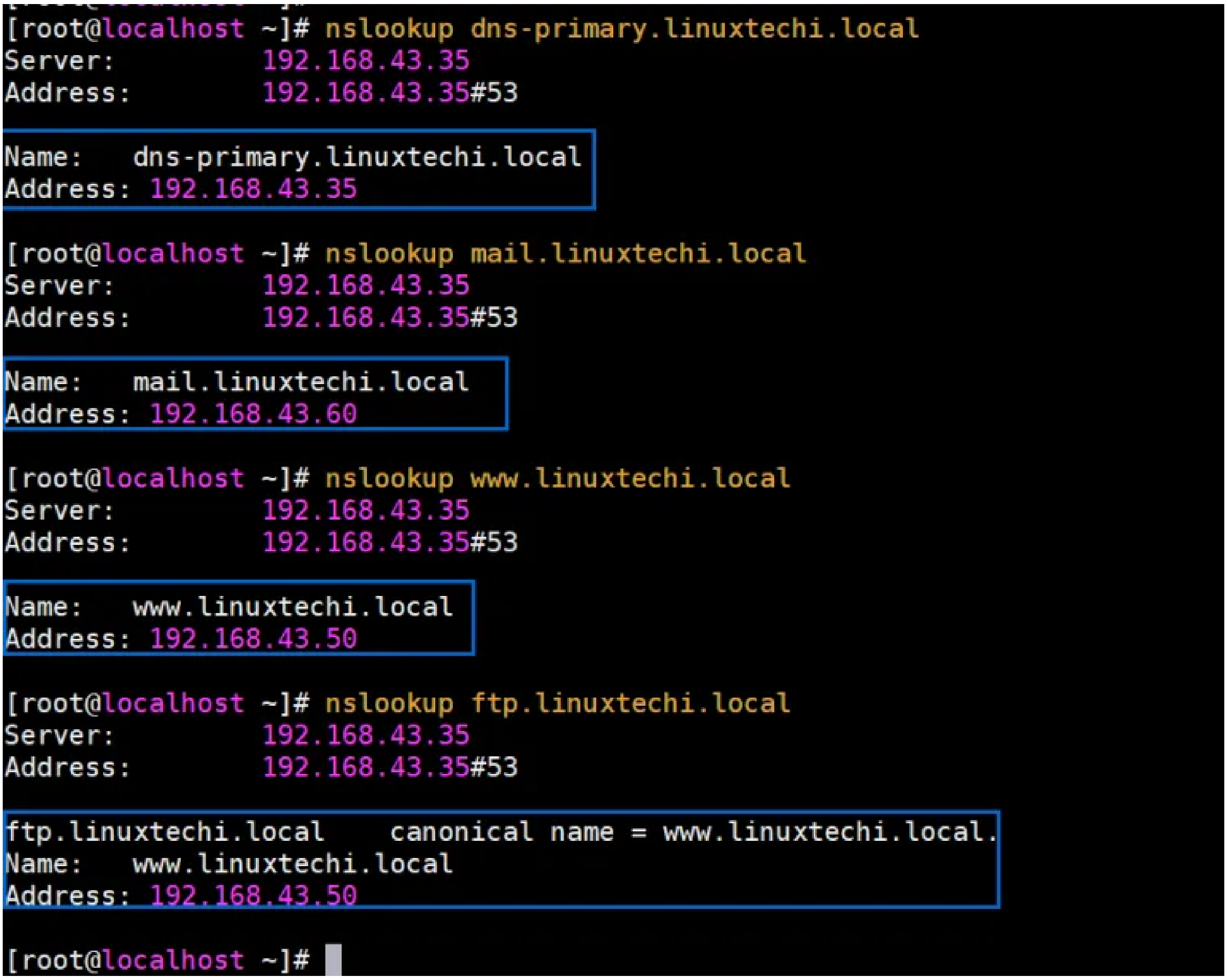
```
TYPE="Ethernet"
PROXY_METHOD="none"
BROWSER_ONLY="no"
BOOTPROTO="dhcp"
DEFROUTE="yes"
IPV4_FAILURE_FATAL="no"
IPV6INIT="yes"
IPV6_AUTOCONF="yes"
IPV6_DEFROUTE="yes"
IPV6_FAILURE_FATAL="no"
IPV6_ADDR_GEN_MODE="stable-privacy"
NAME="enp0s3"
UUID="4fac7fd3-3fd8-4dba-91eb-5518e25e72a4"
DEVICE="enp0s3"
ONBOOT="yes"

DNS=192.168.43.35
~
~
```

Save and close the config file and then restart the network manager service to make above change into the effect,

```
# systemctl restart NetworkManager
```

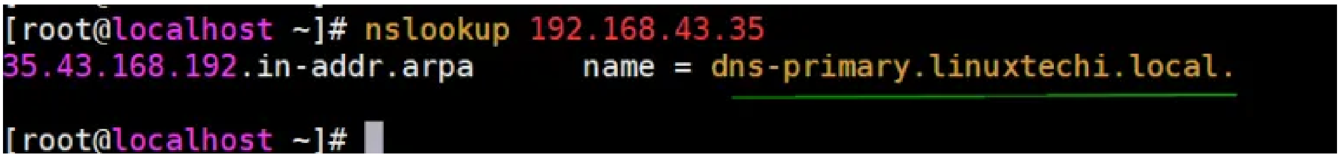
Using the nslookup command test the Bind DNS server as shown:

```
# nslookup dns-primary.linuxtechi.local

# nslookup mail.linuxtechi.local

# nslookup www.linuxtechi.local

# nslookup ftp.linuxtechi.local
```



```
[root@localhost ~]# nslookup dns-primary.linuxtechi.local
Server:         192.168.43.35
Address:        192.168.43.35#53

Name:   dns-primary.linuxtechi.local
Address: 192.168.43.35

[root@localhost ~]# nslookup mail.linuxtechi.local
Server:         192.168.43.35
Address:        192.168.43.35#53

Name:   mail.linuxtechi.local
Address: 192.168.43.60

[root@localhost ~]# nslookup www.linuxtechi.local
Server:         192.168.43.35
Address:        192.168.43.35#53

Name:   www.linuxtechi.local
Address: 192.168.43.50

[root@localhost ~]# nslookup ftp.linuxtechi.local
Server:         192.168.43.35
Address:        192.168.43.35#53

ftp.linuxtechi.local    canonical name = www.linuxtechi.local.
Name:   www.linuxtechi.local
Address: 192.168.43.50

[root@localhost ~]#
```

```
# nslookup 192.168.43.35
```

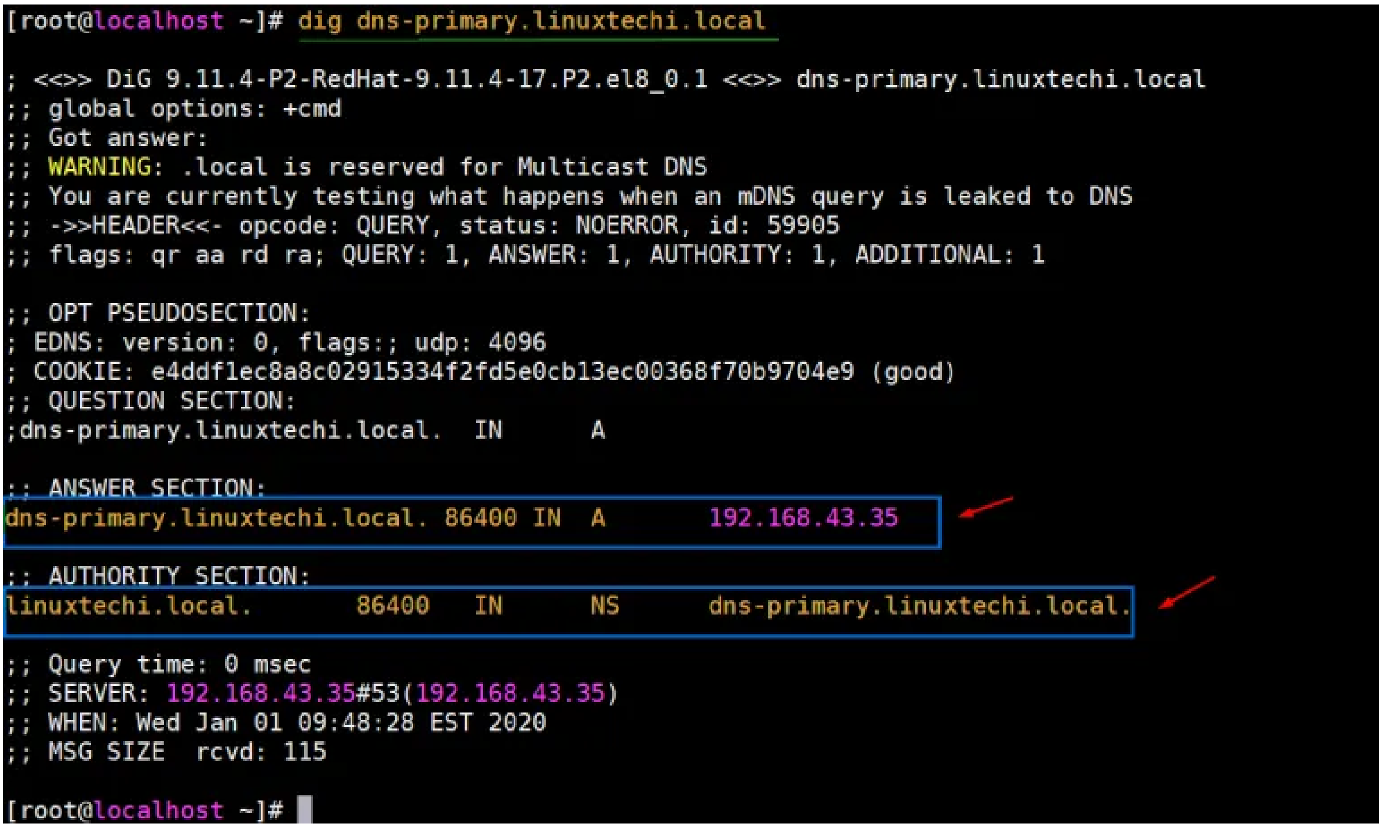


```
[root@localhost ~]# nslookup 192.168.43.35
35.43.168.192.in-addr.arpa      name = dns-primary.linuxtechi.local.

[root@localhost ~]#
```

The output from the nslookup command confirms that the forward DNS lookup is working as expected.

Moreover, you can also use the dig command as shown

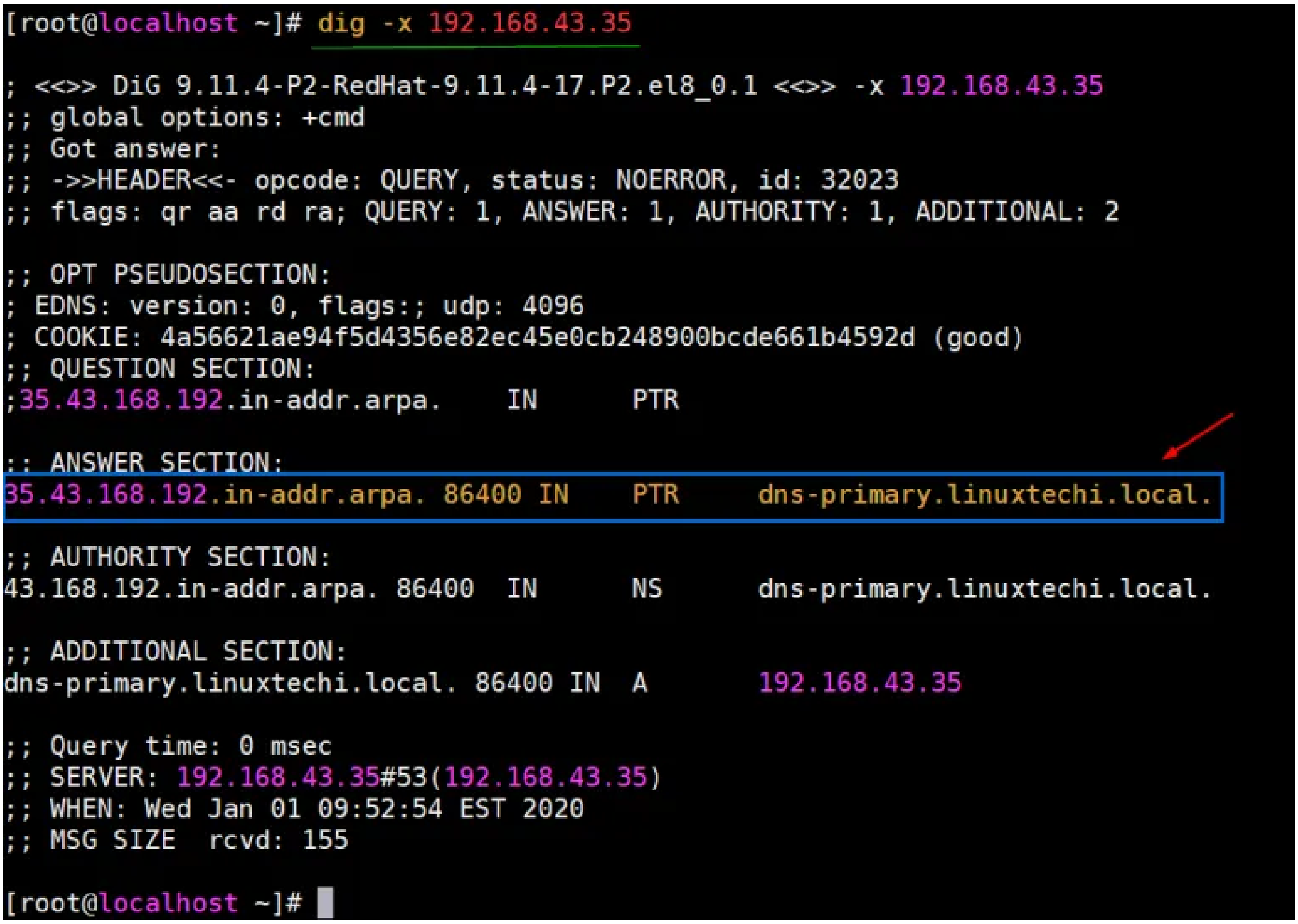
```
# dig dns-primary.linuxtechi.local
```



```
[root@localhost ~]# dig dns-primary.linuxtechi.local

; <<>> DiG 9.11.4-P2-RedHat-9.11.4-17.P2.el8_0.1 <<>> dns-primary.linuxtechi.local
;; global options: +cmd
;; Got answer:
;; WARNING: .local is reserved for Multicast DNS
;; You are currently testing what happens when an mDNS query is leaked to DNS
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 59905
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: e4ddf1ec8a8c02915334f2fd5e0cb13ec00368f70b9704e9 (good)
;; QUESTION SECTION:
;dns-primary.linuxtechi.local.  IN      A

;; ANSWER SECTION:
dns-primary.linuxtechi.local. 86400 IN  A       192.168.43.35

;; AUTHORITY SECTION:
linuxtechi.local.       86400   IN      NS      dns-primary.linuxtechi.local.

;; Query time: 0 msec
;; SERVER: 192.168.43.35#53(192.168.43.35)
;; WHEN: Wed Jan 01 09:48:28 EST 2020
;; MSG SIZE  rcvd: 115

[root@localhost ~]#
```

To perform a reverse DNS lookup, use the dig command as shown:

```
# dig -x 192.168.43.35
```



```
[root@localhost ~]# dig -x 192.168.43.35

; <<>> DiG 9.11.4-P2-RedHat-9.11.4-17.P2.el8_0.1 <<>> -x 192.168.43.35
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 32023
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 4a56621ae94f5d4356e82ec45e0cb248900bcde661b4592d (good)
;; QUESTION SECTION:
;35.43.168.192.in-addr.arpa.    IN      PTR

;; ANSWER SECTION:
35.43.168.192.in-addr.arpa. 86400 IN    PTR     dns-primary.linuxtechi.local.

;; AUTHORITY SECTION:
43.168.192.in-addr.arpa. 86400  IN      NS      dns-primary.linuxtechi.local.

;; ADDITIONAL SECTION:
dns-primary.linuxtechi.local. 86400 IN  A       192.168.43.35

;; Query time: 0 msec
;; SERVER: 192.168.43.35#53(192.168.43.35)
;; WHEN: Wed Jan 01 09:52:54 EST 2020
;; MSG SIZE  rcvd: 155

[root@localhost ~]#
```

Perfect! The reverse DNS lookup is also working as we would expect.

And that wraps up this tutorial. In this guide, we walked you through the installation of a DNS server using Bind on CentOS 8. Give it a try and do not hesitate to send your feedback.

Also Read : **How to Setup NFS Server on CentOS 8 / RHEL 8**

Tags: Bind CentOS 8

14 thoughts on "How to Setup DNS Server (Bind) on CentOS 8 / RHEL8"

RAM

January 4, 2020 at 6:15 am

Thanks.this is very helpful tutorial and will help in clearing DNS related interview questions for Linux admin job.

Reply

STS

January 27, 2020 at 10:02 pm

Thank you for the tutorial. Much appreciated!

Question, we have 3 old bind servers with webmin running on Debian. We would like to setup new CentOS 8 blind servers and sync over the zones. What steps would you recommend?

Reply

TNX

January 29, 2020 at 6:05 pm

Thanks, that helped.
I just had to be aware to edit the named.conf file in /etc/ and the zone files in /var/named/, previously I put the zone files in /etc/named/ and wondered why the service would not start…

Reply

JIGME WANGDI

May 28, 2020 at 11:27 am

i get error while running nslookup 192.168.1.1

Reply

BENSEN GUYER

July 20, 2020 at 3:05 pm

There are a lot of nonsensical tutorials out there. I have to say this one is the most concise yet accurate site I've found. I achieved working DNS without too much issue, but did look for pointers along the way.
I really wish I would have come across this site right away.

Reply

KIRTI MITTAL

October 22, 2020 at 6:33 am

Hi,
I followed each step and configured the DNS-BIND but while doing nslookup from client I am getting below NXDOMAIN error

```
[root@nc0521node02 ~]# nslookup test.nokia.local
Server: 10.10.183.19
Address: 10.10.183.19#53

** server can't find test.nokia.local: NXDOMAIN
```

Can anyone please help to resolve this

Reply

NICK

October 24, 2020 at 7:00 am

Make sure the IP address of the machine that you setup bind on is 10.19.182.19,
Make sure the firewall of the machine that you setup bind on has port 53 open,
double check the entry in your nokia.local.db config file is test IN A 10.10.182.X
Confirm that you restarted named after making the edits to the config files

Try doing the the following test on the machine that is running bind and see if you get a correct response

```
[root@bindMachine ~]# nslookup
> server localhost
> test.nokia.local
```

Reply

NICK

October 24, 2020 at 7:04 am

Sorry formatting on the previous post was terrible…. hopefully this one will be better

Make sure the IP address of the machine that you setup bind on is 10.19.182.19, Make sure the firewall of the machine that you setup bind on has port 53 open, double check the entry in your nokia.local.db config file is test IN A 10.10.182.X Confirm that you restarted named after making the edits to the config files

Try doing the the following test on the machine that is running bind and see if you get a correct response

```
[root@bindMachine ~]# nslookup

> server localhost

> test.nokia.local
```

Reply

NICK

October 24, 2020 at 7:21 am

In step 4 each of the PTR records need to end in a trailing .(dot) otherwise the nslookup for the www or mail will respond with the wrong fqdn along the lines of

```
# nslookup 192.168.43.50
50.43.168.192.in-addr.arpa name = 'www.linuxtechi.local.43.168.192.in-addr.arpa.'
```

Reply

PRADEEP KUMAR

October 25, 2020 at 3:05 am

Thanks Nick,

That's was the typo, I have correct it now.

Reply

VICTOR

November 10, 2020 at 8:56 pm

if the web and mail server would be in the same pc of the dns would they have the same ip?
www IN A 192.168.43.50 –>.35
mail IN A 192.168.43.6 –>.35

how would the reverse file be?

I have that doubt

Reply

ARUN KUMAR
December 30, 2020 at 7:02 pm

but question is how other hosts in network will use this DNS server for name resolution setting up just DN S is of no use till we use for name resolution for servers in networks how can we do that

Reply

PRADEEP KUMAR
December 31, 2020 at 3:47 am

Hi Arun,

To use the DNS server on client machines, you have to add DNS server IP address (for linux machine you can add it in resolv.conf & on windows machine edit network settings for active NIC and add DNS server ip there)

Reply

BOB
January 5, 2021 at 5:01 pm

I have been working for come next month for 3 years trying to setup a home server to handle our own websites . So on a spur of the moment type deal i dumped centos 7 and installed centos 8 about 6 times due to errors of the AppStream but finally got it. After words i went on the hunt for help on setting up centos 8 server and came across your tutitorial and it is simply the best that i found in the 3 years i have been working on this project. most everyone else leaves out details that are vital and you have to go on a hunt to try and to find the missing pieces and then you loose track of where you where. thank you for including everything i needed to set my server up. you are greatly appreciated by me for doing such a wonderful job on sharing the right information to help someone like me. again i can’t thank you enough.

Reply

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

Website

□ I'm not a robot

reCAPTCHA
Privacy - Terms

POST COMMENT

PREVIOUS

NEXT

How to Boot CentOS 8 / RHEL 8 Server in Single User Mode

How to Install Prometheus on CentOS 8 / RHEL 8

Search …

Recent Posts

How to Repair Corrupted XFS Filesystem with xfs_repair

How to Install Docker and Docker-Compose on Rocky Linux 8

How to Extend XFS Root Partition without LVM in Linux

Top 7 PDF Editors for Ubuntu Linux

Top 6 Screenshot Tools for Ubuntu / Linux Mint / Debian

How to Install Ansible AWX on Kubernetes Minikube

Top 8 Linux Distributions for Developers in 2021

Easy Guide to Migrate from CentOS 8 to Rocky Linux 8

How to Install Ansible on Ubuntu 20.04 LTS / 21.04

How to Install Rocky Linux 8.4 Step by Step with Screenshots