How to Set Up Private DNS Servers with BIND on CentOS 8

January 8, 2021 by Jeff Wilson

BIND also know as the “Berkeley Internet Name Domain” is one of the most popular Domain Name System (DNS) now a day. It an open-source and provides DNS services on Linux operating systems. Generally, it helps you to resolve a fully qualified domain name into an IP address or IP address to a domain name. It can be used as an authoritative name server and provides several features like load balancing, dynamic update, split DNS, etc.

In this tutorial, we will show you how to set up a private DNS server with BIND on CentOS 8.

Table of Contents

Prerequisites

- A CentOS 8 VPS (we’ll be using our SSD 2 VPS plan)
- Access to the root user account (or access to an admin account with root privileges)

For the purpose of this tutorial, we will use the following setup:

Hostname : ns1.rosehosting.local
IP Address : 192.168.1.100
Local Network : 192.168.1.0/24

Step 1: Log in to the Server & Update the Server OS Packages

First, log in to your CentOS 8 server via SSH as the root user:

```
ssh root@IP_Address -p Port_number
```

You will need to replace 'IP_Address' and 'Port_number' with your server's respective IP address and SSH port number. Additionally, replace 'root' with the username of the admin account if necessary.

Before starting, you have to make sure that all CentOS packages installed on the server are up to date. You can do this by running the following commands:

```
dnf update -y
```

Step 2: Install BIND DNS Server

By default, the bind package is available in the CentOS 8 standard repository. You can install it by running the following command:

```
dnf install bind bind-utils -y
```

Once the BIND is installed, start the BIND service and enable it to start at system reboot:

```
systemctl start named
systemctl enable named
```

Step 3: Configure BIND DNS Server

By default, the BIND server is listening on localhost only. So you will need to configure it to listen on all network interfaces. You can configure it by editing the file /etc/named.conf:

```
nano /etc/named.conf
```

Comment out the following lines:

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

Change the following line to allow query for your local network:

```
allow-query     { localhost;192.168.1.0/24; };
```

Save and close the file when you are finished.

Step 4: Create Forward and Reverse DNS Zone

A Forward Zone is used to resolve the hostname to IP address while a Reverse Zone is used to resolve the IP address to hostname. Generally, all normal DNS queries are forward lookup queries. You can define the forward and reverse lookup zones in the

/etc/named.conf file.

Edit the /etc/named.conf file with the following command:

```
nano /etc/named.conf
```

Add the following lines at the end of the file

```
//Forward Zone
zone "rosehosting.local" IN {

            type master;
            file "rosehosting.local.db";
            allow-update { none; };

};

//Reverse Zone
zone "1.168.192.in-addr.arpa" IN {
            type master;
            file "192.168.1.db";
            allow-update { none; };

};
```

Save and close the file when you are finished.

Step 5: Create Forward and Reverse Zone Files

Next, you will need to create a forward and reverse zone files defined in the previous step. By default, all zone lookup files are located inside /var/named directory.

First, create a forward zone file with the following command:

```
nano /var/named/rosehosting.local.db
```

Add the following lines:

```
$TTL 86400
@   IN  SOA     ns1.rosehosting.local. root.rosehosting.local. (
                                            3           ;Serial
                                            3600        ;Refresh
                                            1800        ;Retry
                                            604800      ;Expire
                                            86400       ;Minimum
TTL
)

;Name Server Information
@       IN  NS      ns1.rosehosting.local.

;IP address of Name Server
ns1       IN  A       192.168.1.100

;A - Record HostName To Ip Address
www     IN  A       192.168.1.101

;CNAME record
ftp     IN CNAME        www.rosehosting.local.
```

Save and close the file then create a reverse zone file with the following command:

```
nano /var/named/192.168.1.db
```

Add the following lines:

```
$TTL 86400
@   IN  SOA     ns1.rosehosting.local. root.rosehosting.local. (
                                        3           ;Serial
                                        3600        ;Refresh
                                        1800        ;Retry
                                        604800      ;Expire
                                        86400       ;Minimum TTL
)

;Name Server Information
@         IN      NS         ns1.rosehosting.local.

;Reverse lookup for Name Server
100       IN  PTR     ns1.rosehosting.local.

;PTR Record IP address to HostName
101      IN  PTR     www.rosehosting.local.
```

Save and close the file when you are finished.

Step 6: Verify DNS Configuration

After configuring all zone files, you will need to verify the configuration files.

First, validate the main configuration file with the following command:

```
named-checkconf /etc/named.conf
```

If everything is fine, you don't see any error.

Next, verify the forward zone file with the following command:

```
named-checkzone rosehosting.local /var/named/rosehosting.local.db
```

You should get the following output:

```
zone rosehosting.local/IN: loaded serial 3
OK
```

Next, verify the reverse zone file with the following command:

```
named-checkzone 1.168.192.in-addr.arpa /var/named/192.168.1.db
```

Yu should get the following output:

```
zone 1.168.192.in-addr.arpa/IN: loaded serial 3
OK
```

Finally, restart the BIND service to apply the changes:

```
systemctl restart named
```

Step 7: Configure Firewall

Next, you will need to create a firewall rule for port 53 to allow DNS queries from client machines. You can create it with the following command:

```
firewall-cmd --permanent --add-port=53/udp
```

Next, reload the filewall service to apply the changes:

```
firewall-cmd --reload
```

Step 8: Verify DNS Server

At this point, the BIND DNS server is installed and configured. It's time to check whether it is working or not.

First, edit your /etc/resolv.conf file and add your DNS server IP:

```
nano /etc/resolv.conf
```

Add the following line at the beginning of the file:

```
nameserver 192.168.1.100
```

Save and close the file then verify the forward lookup using the dig command:

```
dig www.rosehosting.local
```

Or

```
dig ns1.rosehosting.local
```

If everything is fine, you should get the following response:

```
; <<>> DiG 9.11.20-RedHat-9.11.20-5.el8 <<>> www.rosehosting.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: 52518
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1,
ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: cd9d365f1f02621aa9c8753c5fd47154db8cae737b9ca09f (good)
;; QUESTION SECTION:
;www.rosehosting.local.        IN  A

;; ANSWER SECTION:
www.rosehosting.local.    86400   IN  A   192.168.1.101

;; AUTHORITY SECTION:
rosehosting.local.    86400   IN  NS  ns1.rosehosting.local.

;; ADDITIONAL SECTION:
ns1.rosehosting.local.    86400   IN  A   192.168.1.100

;; Query time: 0 msec
;; SERVER: 192.168.1.100#53(192.168.1.100)
;; WHEN: Sat Dec 12 02:29:24 EST 2020
;; MSG SIZE  rcvd: 128
```

Next, verify the reverse lookup with the following command:

```
dig -x 192.168.1.100
```

You should get the following response:

```
; <<>> DiG 9.11.20-RedHat-9.11.20-5.el8 <<>> -x 192.168.1.100
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 30878
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1,
ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 18a66bab478cf57219e6c17c5fd471671887a1dd983fef57 (good)
;; QUESTION SECTION:
;100.1.168.192.in-addr.arpa.    IN  PTR

;; ANSWER SECTION:
100.1.168.192.in-addr.arpa. 86400 IN    PTR ns1.rosehosting.local.

;; AUTHORITY SECTION:
1.168.192.in-addr.arpa.    86400   IN  NS  ns1.rosehosting.local.

;; ADDITIONAL SECTION:
ns1.rosehosting.local.    86400   IN  A   192.168.1.100

;; Query time: 0 msec
;; SERVER: 192.168.1.100#53(192.168.1.100)
;; WHEN: Sat Dec 12 02:29:43 EST 2020
;; MSG SIZE  rcvd: 148
```

Congratulations! you have successfully set up a private DNS server with BIND on CentOS 8.

Of course, you don't have to do any of this if you use one of our Linux VPS Hosting services, in which case you can simply ask our expert Linux admins to setup this for you. They are available 24×7 and will take care of your request immediately.

PS. If you liked this post please share it with your friends on the social networks using the buttons on the left or simply leave a reply below. Thanks.

CentOS, Networking and Domains

< How to Install Apache Cassandra on CentOS 8

> How to Install CouchDB on Ubuntu 20.04

2 thoughts on "How to Set Up Private DNS Servers with BIND on CentOS 8"

Jezrel Jane
May 31, 2021 | Reply

Thank you for this guide!

Alejandro HB
July 21, 2021 | Reply

Thank you ! very good.

Leave a Comment

Name *

Email *

☐ Save my name, email, and website in this browser for the next time I comment.

To prove you are human please solve the following *

5 × 5 =

☐ Yes, add me to your new blog post notifications list

Post Comment