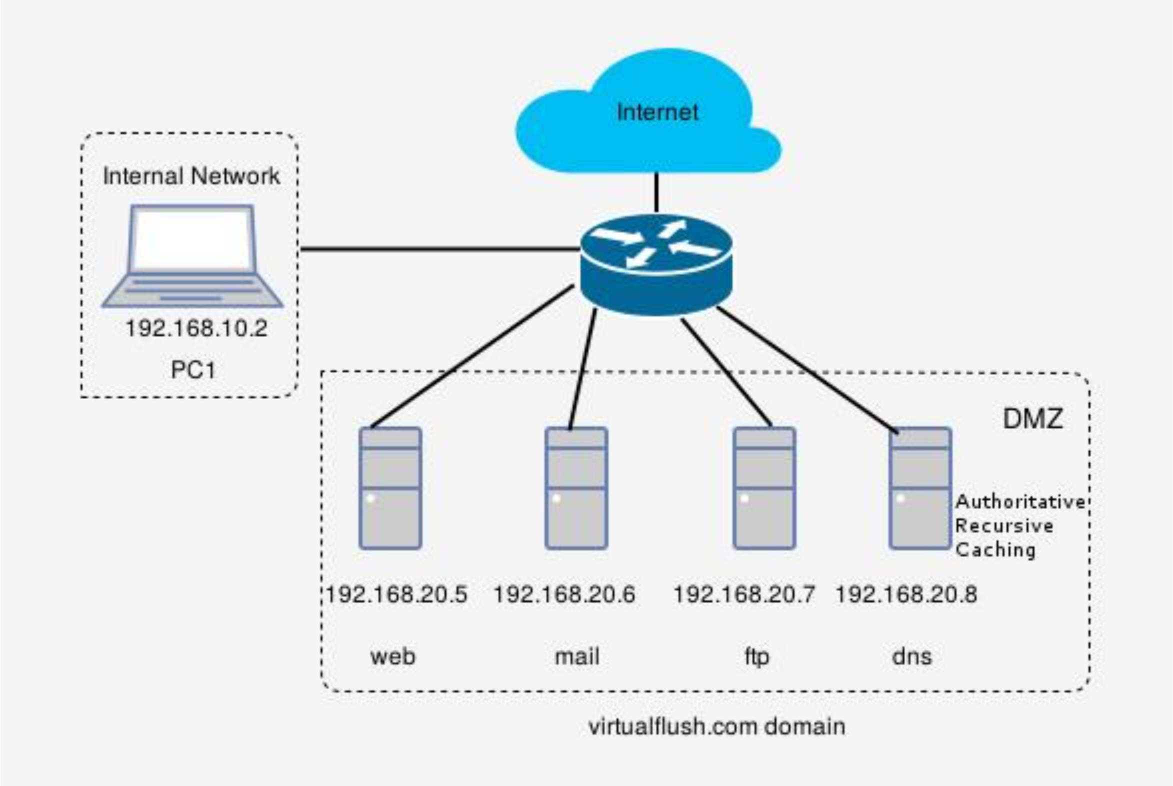
How To Install and Configure DNS Server on Linux Debian

A DNS server is used to resolve an IP address to a hostname and vice versa.
The most popular name server on Linux is BIND, stands for Berkley Internet Naming Daemon.

This tutorial explains how to install and configure Bind on Debian 8.4, using the diagram below as an example for installation.



Step 1. Install BIND

BIND is available from the default Debian repositories, so we ca install the software using *apt-get* command-line tool.

```
dns# apt-get install bind9
```

Check to see if Bind is running:

```
dns# service bind9 status
```

Step 2. Create Zone Files on Master Server

Our DNS server will be authoritative for *virtualflush.com* domain. An authoritative name server can either be a primary/master or a secondary/slave server.

A slave DNS server will respond to requests if the primary server becomes unavailable and it is highly recommended to set up one ore more slave servers, but it is not the purpose of this tutorial because we have only one DNS server.

Bind configuration files are kept in a */etc/bind* directory. Let's change directory to */etc/bind* and create a new folder where forward and reverse zone files will reside.

```
dns# cd /etc/bind

dns# mkdir master_zones

dns# cd master_zones
```

Create forward zone:

```
dns# nano db.virtualflush.com
```

The file should look like this:

```
; BIND data file for virtualflush.com

$TTL    86400

@       IN      SOA     dns.virtualflush.com. root.virtualflush.com. (

                          2016051001        ; Serial

                          3h                ; Refresh after 3 hours

                          1h                ; Retry after 1 hour

                          1w                ; Expire after 1 week

                          3h )              ; Negative caching TTL of 3 hours

;

                NS          dns.virtualflush.com.

                MX    10    mail.virtualflush.com.
```

```
web             A       192.168.20.5
mail            A       192.168.20.6
ftp             A       192.168.20.7
dns             A       192.168.20.8

www             CNAME   web
```

We have the forward zone configured and now we must to create the reverse zone file (reverse DNS is the opposite of forward DNS).

Note that you can manage reverse DNS for a subnet only if your Internet Service Provider has been delegated authority to your nameservers, otherwise you must contact the support department of your ISP.

In the reverse zone file we will define PTR records for reverse DNS lookups.

```
dns# nano db.192.168.20
```

The file should look like this:

```
; BIND reverse data file for 20.168.192.in-addr.arpa
$TTL   86400
@      IN      SOA     dns.virtualflush.com. root.virtualflush.com. (
                          2016051001        ; Serial
                          3h                ; Refresh after 3 hours
                          1h                ; Retry after 1 hour
                          1w                ; Expire after 1 week
                          3h )              ; Negative caching TTL of 3 hours
;
                          NS      dns.virtualflush.com.

5                         PTR     web.virtualflush.com.
```

```
6                         PTR     mail.virtualflush.com.
7                         PTR     ftp.virtualflush.com.
8                         PTR     dns.virtualflush.com.
```

Insert both zone file names into *named.conf.local* file:

```
dns# nano /etc/bind/named.conf.local

zone "virtualflush.com" {
       type master;
       file "/etc/bind/master_zones/db.virtualflush.com";
};

zone "20.168.192.in-addr.arpa" {
       type master;
       file "/etc/bind/master_zones/db.192.168.20";
};
```

Step 3. Edit Options File

We will configure this server to act as a caching DNS server and to performs recursive queries from our “Internal Network” clients – 192.168.10.0/24.

A caching DNS server performs recursive queries and stores the answer in its cache for a specific period (TTL value from zone records).

Just edit the *named.conf.options* file to look like this:

```
dns# nano /etc/bind/named.conf.options

options {
        directory "/var/cache/bind";
        dnssec-validation auto;

        auth-nxdomain no;    # conform to RFC1035
        listen-on-v6 { any; };

        recursion yes;
        allow-recursion { 192.168.10.0/24; };
};
```

Note: If you want to forward requests to other DNS server, you can configure your server as a forwarding DNS server.
A forwarding DNS server is like a caching server but it does not performs recursive queries itself. Instead, it forwards all requests to other resolving server and then caches the results.
You can easily set up a forwarding DNS by adding the *forwarders* directive inside *named.conf.local* file.

The configuration file should look like this:

```
options {
        directory "/var/cache/bind";
        dnssec-validation auto;

        auth-nxdomain no;    # conform to RFC1035
        listen-on-v6 { any; };

        recursion yes;
        allow-recursion { 192.168.10.0/24; };

        forwarders {
           //Google’s public DNS servers
           8.8.8.8;
           8.8.4.4;
        };
};
```

Step 4. Test Bind Configuration Files

Check for errors in the configuration files by running the following command:

```
dns# named-checkconf
```

If there are errors found in configuration file, they will be displayed on screen, otherwise the above command will not display anything.

Now restart Bind service:

```
dns# service bind9 restart
```

Step 5. Test DNS Resolution

Test forward lookup with the *dig* command:

```
PC1# dig web.virtualflush.com @192.168.20.8 +short
192.168.20.5
PC1# dig mx virtualflush.com @192.168.20.8 +short
```

```
10 mail.virtualflush.com.
```

Test reverse lookup:

```
PC1# dig -x 192.168.20.5  @192.168.20.8 +short

web.virtualflush.com.
```

Finally, we need to configure client machine to use the DNS sever. On PC1, edit */etc/resolv.conf* file and add a *nameserver* line like this:

```
PC1# cat /etc/resolv.conf



nameserver 192.168.20.8
```