dns on ubuntu

We're going to show you how to setup DNS BIND on Ubuntu Linux. This was tested on Ubuntu 20.04. We're going to show you how to setup a primary and secondary server. You can just skip the secondary if you only need one.

Install and Initial Setup

Run the steps in this section on both the primary and secondary DNS server.
Install BIND:

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

IPv4 Mode - Set this up unless you have IPv6.

```
sudo nano /etc/default/bind9
```

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

Restart BIND:

```
sudo systemctl restart bind9
```

Create a directory for the zone files:

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

Make sure that the local firewall isn't blocking DNS:

```
sudo ufw allow Bind9
```

Primary DNS Server

Add IPs to the trusted ACL. Configure options including enabling recurstion from trusted clients. Tell the server which interface / IP to listen on. Turn off zone transfers by default and add forwarders. In this case we are using Google's DNS servers as our forwarders.
The config file should look similar to this:

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

acl "trusted" {
192.168.3.3; # ns1 - can be set to localhost
192.168.3.4; # ns2
192.168.3.5; # host1
192.168.3.6; # host2
};
options {
directory "/var/cache/bind";

recursion yes; # enables resursive queries
allow-recursion { trusted; }; # allow trusted clients to perform
recursive queries
listen-on { 192.168.3.3; }; # listen on this IP ( ex: private network
)
allow-transfer { none; }; # by default disable zone transfers

forwarders {
8.8.8.8;
8.8.4.4;
};
dnssec-validation auto;
};
```

Add zones to the following file. It will also specify where the zonefiles are located.

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


zone "example.org" {
type master;
file "/etc/bind/zones/db.example.org"; # zone file path
allow-transfer { 192.168.3.4; }; # ns2 private IP address - secondary
};
```

```
zone "168.192.in-addr.arpa" {
type master;
file "/etc/bind/zones/db.192.168"; # 192.168.0.0/24 subnet
allow-transfer { 192.168.3.4; }; # ns2 private IP address - secondary
};
```

Forward Zone Files

Forward zone files will contain A records. These resolve hostnames to IP addresses. This is also where you will define CNAMES and NS records.

IMPORTANT - Always update the serial number when editing a zone file. Restart the daemon to make the changes effective.

The serial number will be defined on a line that looks like this. It is important to update this every time you make an update to a zone file.

```
3 ; Serial
```

Set the names to match our domain. Define name servers and A records. The file should look similar to this.

```
sudo nano /etc/bind/zones/db.example.org
```

```
$TTL 604800
@ IN SOA ns1.example.org. admin.example.org. (
4 ; Serial
604800 ; Refresh
86400 ; Retry
2419200 ; Expire
604800 ) ; Negative Cache TTL
;


; name servers - NS records
IN NS ns1.example.org.
IN NS ns2.example.org.


; name server - A records
ns1.example.org. IN A 192.168.3.3
ns2.example.org. IN A 192.168.3.4
```

```
; 192.168.3.0/24 - A records
host1.example.org. IN A 192.168.3.5
host2.example.org. IN A 192.168.3.6
host3.example.org. IN CNAME host2.example.org.
```

Reverse Zone Files

Reverse zone files will contain PTR records. These resolve IP addressses to hostnames.
IMPORTANT - Always update the serial number when editing a zone file. Restart the daemon to make the changes effective.
Update this with our domain name. Define name servers and PTR records. The file should look similar to this.

```
sudo nano /etc/bind/zones/db.192.168
```

```
$TTL 604800
@ IN SOA ns1.example.org. admin.example.org. (
3 ; Serial
604800 ; Refresh
86400 ; Retry
2419200 ; Expire
604800 ) ; Negative Cache TTL
;

; name servers - NS records
IN NS ns1.example.org.
IN NS ns2.example.org.


; PTR Records
3.3 IN PTR ns1.example.org. ; 192.168.3.3
4.3 IN PTR ns2.example.org. ; 192.168.3.4
5.3 IN PTR host1.example.org. ; 192.168.3.5
6.3 IN PTR host2.example.org. ; 192.168.3.6
```

More

Verify syntax of config files:

```
sudo named-checkconf
```

Verify syntax of zone files:

```
sudo named-checkzone example.org /etc/bind/zones/db.example.org
```

```
sudo named-checkzone 168.192.in-addr.arpa /etc/bind/zones/db.192.168
```

Restart BIND:

```
sudo systemctl restart bind9
```

Secondary DNS Server

Edit the named.conf.options file.

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

It will include basically the same changes as primary server except that the line defining which interface to listen on will be different:

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

```
listen-on { 192.168.3.4; };
```

This is a bit different from the version on the primary server. It will be configured to point to the primary. It also sets the type to 'slave' instead of 'master'.

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

```
zone "example.org" {
type slave;
file "db.example.org";
masters { 192.168.3.3; }; # ns1
};

zone "168.192.in-addr.arpa" {
type slave;
file "db.192.168";
masters { 192.168.3.3; }; # ns1
};
```

Check the config syntax and restart BIND:

```
sudo named-checkconf
sudo systemctl restart bind9
```

Client Setup

Note that the client configuration on Windows and Mac OS will be quite a bit different. Other Linux distros will be different as well.

Ubuntu 18.04 / 20.04 Client

Assuming you are using the Netplan network manager and that you want to configure it using the command line, you can do the following. You will probably need to change the interface name. You might also want to just add this to your regular network config if you have a static IP configured.

```
sudo nano /etc/netplan/00-private-nameservers.yaml

network:
version: 2
ethernets:
eth1: # Private network interface
nameservers:
addresses:
- 192.168.3.3 # Private IP for ns1
- 192.168.3.4 # Private IP for ns2
search: [ example.org ] # DNS zone
```

Tell Netplan to attempt to load the configuration:

```
sudo netplan try
```

Verify if the DNS configuration has been applied.

```
sudo systemd-resolve --status
```

Known Issue with /etc/resolv.conf

There is a known issue with systemd see HERE.

The file /etc/resolv.conf is meant to be automatically updated by man:systemd-resolved. This doesn't happen. The file /run/systemd/resolve/resolv.conf is updated sucessfully though. We can work around this by removing the first file and replacing it with a link to the second like this.

```
sudo rm -f /etc/resolv.conf
sudo ln -s /run/systemd/resolve/resolv.conf /etc/resolv.conf
```

DNS Utils Install

You should have nslookup and dig by default on the server version of Ubuntu and probably on the client version as well. If you don't have them installed for whatever reason you can use the following command.

```
apt-get install dnsutils
```

Ubuntu 16.04 Client

```
sudo nano /etc/network/interfaces
```

```
dns-nameservers 192.168.3.3 192.168.3.4 8.8.8.8
dns-search example.org
```

Restart networking (swap eth0 for the correct device on your system).

```
sudo ifdown --force eth0 && sudo ip addr flush dev eth0 && sudo ifup
--force eth0
```

Verify that your DNS servers and domain are visible here:

```
cat /etc/resolv.conf
```

Testing

Do a forward lookup using the host command. You should be able to lookup the hostname by itself as well as using the FQDN.

```
host host1
host host2
host host1.example.org
host host2.example.org
```

Do the same test with the nslookup command.

```
nslookup host1
nslookup host2
nslookup host1.example.org
nslookup host2.example.org
```

You can also test with the dig command:

```
dig host1.example.org
dig host2.example.org
```

Do a reverse lookup using the host command, the nslookup command, and the dig command.

```
host 192.168.3.5
nslookup 192.168.3.5
dig -x 192.168.3.5
```

Configs Updates

Anytime you update the config files run this command to validate:

```
sudo named-checkconf
```

Anytime you update a zone file make sure you:

- increment the serial number
- run the named-checkzone command

```
sudo named-checkzone example.org /etc/bind/zones/db.example.org
sudo named-checkzone 168.192.in-addr.arpa /etc/bind/zones/db.192.168
```

After making a change and performing the above checks restart bind:

```
sudo systemctl reload bind9
```