Practical 5:- Install DNS Server BIND, Configure DNS server which resolves domain name or IP address, Install BIND 9, Configure BIND, Limit ranges you allow to access if needed.

Solution:-

Step 1:- Installation of Bind 9 on Ubuntu System:

We need to install 'bind9 bind9utils bind9-doc dnsutils' to install BIND 9 & related tools. Open your terminal & execute the following command,

 # sudo apt-get install bind9 bind9utils bind9-doc dnsutils

```
rootclient@ubuntu:~$ sudo apt-get install bind9 bind9utils bind9-doc dnsutils
Reading package lists... Done
Building dependency tree
Reading state information... Done
dnsutils is already the newest version (1:9.11.3+dfsg-1ubuntu1.9).
dnsutils set to manually installed.
Suggested packages:
  resolvconf python-ply-doc
The following NEW packages will be installed:
  bind9 bind9-doc bind9utils python3-ply
0 upgraded, 4 newly installed, 0 to remove and 7 not upgraded.
Need to get 893 kB of archives.
After this operation, 5,315 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu bionic/main amd64 python3-ply all 3.11
-1 [46.6 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu bionic-updates/main amd64 bind9utils a
md64 1:9.11.3+dfsg-1ubuntu1.9 [216 kB]
```

Step 2:-Set BIND to IPv4 Mode

Set BIND to IPv4 mode, we will do that by editing the "/etc/default/bind9" file and adding "-4" to the OPTIONS variable:-

 # sudo nano /etc/default/bind9

```
  GNU nano 2.9.3                    /etc/default/bind9

#
# run resolvconf?
RESOLVCONF=no

# startup options for the server
OPTIONS="-4 -u bind"
```

Now let's configure ns1,our primary DNS server.

Step 3:- Configuration of Primary server

Once all the packages have been installed, we will move into the configuration part. All configuration files for BIND are located in folder **'/etc/bind'**.

```
rootclient@ubuntu:~$ ls /etc/bind
bind.keys   db.empty    named.conf.default-zones  zones.rfc1918
db.0        db.local    named.conf.local
db.127      db.root     named.conf.options
db.255      named.conf  rndc.key
```

One of the important configuration file for bind is **"/etc/bind/named.conf.options",** from this file we can set the followings parameters:

- Allow Query to your dns from your private network (As the name suggests only the systems from your private network can query dns sever for name to ip translation and vice-versa)
- Allow recursive query
- Specify the DNS port (53)
- Forwarders (DNS query will be forwarded to the forwarders when your local DNS server is unable to resolve query)

As per my private network settings, I have specified the following parameters:

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

```
 GNU nano 2.9.3                                  /etc/bind/named.conf.options

acl "trusted"
{
192.168.171.132;
192.168.171.141;
192.168.171.151;
192.168.171.121;

};
options {
        directory "/var/cache/bind";

        // If there is a firewall between you and nameservers you want
        // to talk to, you may need to fix the firewall to allow multiple
        // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

        // If your ISP provided one or more IP addresses for stable
        // nameservers, you probably want to use them as forwarders.
        // Uncomment the following block, and insert the addresses replacing
        // the all-0's placehoer.
         forwarders {8.8.8.8;8.8.4.4;};

        //========================================================================
        // If BIND logs error messages about the root key being expired,
        // you will need to update your keys.  See https://www.isc.org/bind-keys
        //========================================================================
        dnssec-validation auto;
        allow-recursion { trusted;};
        listen-on port 53 {localhost;192.168.171.132/24; };
        allow-query{trusted;};

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

Next Important Configuration file is "**/etc/bind/named.conf.local**", in this file we will define the zone files for our domain, edit the file add the following entries:

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

```
 GNU nano 2.9.3                                    /etc/bind/named.conf.local

//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";
zone  "PrimaryS.local"
{
 type master;
 file "/etc/bind/zone/forward.PrimaryS.local";
 allow-transfer {192.168.171.121;};
};

zone "0.168.192.in-addr.arpa"
{
 type master;
 file "/etc/bind/zone/reverse.PrimaryS.local";
 allow-transfer {192.168.171.121; };

};
```

Save the file & exit. Here we have mentioned locations for our forward lookup zone file & reverse lookup zone files. Next we will creat192.168.171.132 the mentioned forward & reverse zone files.

Step 3.1:- Create the Mentioned Forward & Reverse Zone Files

Firstly create the forward lookup zone file, Sample zone files (db.local) are already there in '**/etc/bind'** folder, we can use and copy sample zone file,

 # sudo cp db.local forward.PrimaryS.local

Your forward lookup file should look like something below:

```
 GNU nano 2.9.3                          /etc/bind/forward.PrimaryS.local

;
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     primary.PrimaryS.local. admin.PrimaryS.local. (
                              3         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;name servers - A records
IN      NS      Primary.PrimaryS.local.
;
primary.PrimaryS.local    IN    A       192.168.171.132
; 192.168.171.0/24 -A records
host1.PrimaryS.local    IN    A   192.168.141
;
PrimaryS.local  IN       MX  10   mail.PrimaryS.local.
;
www     IN       A       192.168.171.141
;
mail    IN       A       192.168.171.151
;
ftp      IN      CNAME www.PrimaryS.local.
```

Here, we have added information regarding our DNS server & have also added A records for couple of servers, also added record for a mail server & **CNAME record** for ftp server. Make sure you edit this file to suit your network.

Next we will create a **reverse lookup zone file** at the same location, sample reverse lookup zone file is present at '**/etc/bind**'folder.
 # sudo cp db.127 reverse.PrimaryS.local
 # sudo nano reverse.PrimaryS.local
Your Reverse Zone Lookup file should look like below:

```
 GNU nano 2.9.3                    /etc/bind/reverse.PrimaryS.local

;
; BIND reverse data file for local loopback interface
;
$TTL    604800
@       IN      SOA     primary.PrimaryS.local. admin.PrimaryS.local.               (
                              21         ; Serial
                          604800         ; Refresh
                           86400         ; Retry
                         2419200         ; Expire
                          604800 )       ; Negative Cache TTL
;
        IN      NS      primary.PrimaryS.local.
;
primary.PrimaryS.local.   IN    A     192.168.171.132 ;
171.132 IN      PTR     primary.PrimaryS.local. ;
171.141    IN     PTR     www.PrimaryS.local. ;
171.151     IN    PTR     mail.PrimaryS.local. ;
```

Save file & exit. Now all we have to do is to restart the BIND service to implement the changes made,

In case OS firewall is running on your bind server ten execute the below command to allow 53 port

```
rootclient@ubuntu:~$ sudo systemctl restart bind9
[sudo] password for rootclient:
rootclient@ubuntu:~$ sudo systemctl enable bind9
Synchronizing state of bind9.service with SysV service script with /lib/systemd/
systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable bind9
rootclient@ubuntu:~$ sudo ufw allow 53
Rules updated
Rules updated (v6)
```

Step 3.2:- Validating Syntax of bind9 configuration and Zone files
If you want to cross verify the syntax of your bind 9 configuration file (named.conf.local). Use the command "**named-checkconf**", example is shown below:

```
rootclient@ubuntu:~$ sudo named-checkconf /etc/bind/named.conf.local
/etc/bind/named.conf.local:12: missing ';' before '}'
rootclient@ubuntu:~$ sudo nano /etc/bind/named.conf.local
rootclient@ubuntu:~$ sudo named-checkconf /etc/bind/named.conf.local
```

If there is some syntax error in your bind configuration file, then make changes. After that again run the named-checkconf command.

If there is no syntax error in your bind configuration file, then it should return to shell without showing any errors.

To cross verify the syntax your forward and reverse lookup zone files , use the command "**named-checkzone**", example is shown below:

```
rootclient@ubuntu:~$ sudo named-checkzone PrimaryS.local /etc/bind/forward.PrimaryS.local
zone PrimaryS.local/IN: loaded serial 5
OK
rootclient@ubuntu:~$ sudo named-checkzone PrimaryS.local /etc/bind/reverse.PrimaryS.local
zone PrimaryS.local/IN: loaded serial 21
OK
```

Step 4:- Testing the DNS server with dig & nslookup

To test out our BIND 9 DNS server, we will use another Ubuntu machine & will change its DNS to point out our DNS server. To change the DNS server, open '**/etc/resol.conf**'& make the following DNS entry,

#sudo nano /etc/resolv.conf

save the file & exit. We now have our client ready with DNS pointing to our server. We will now use a CLI tool called '**dig**' command , which is used to get find out DNS & its related information. Execute the following command from terminal,

 # sudo dig primary.PrimaryS.local

& we should get the following output from the command,

```
rootclient@ubuntu:~$ dig primary.PrimaryS.local

; <<>> DiG 9.11.3-1ubuntu1.9-Ubuntu <<>> primary.PrimaryS.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: SERVFAIL, id: 28098
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;primary.PrimaryS.local.                IN      A

;; Query time: 6 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Sat Sep 21 03:03:35 PDT 2019
;; MSG SIZE  rcvd: 51
```

This output shows that our DNS is working fine.