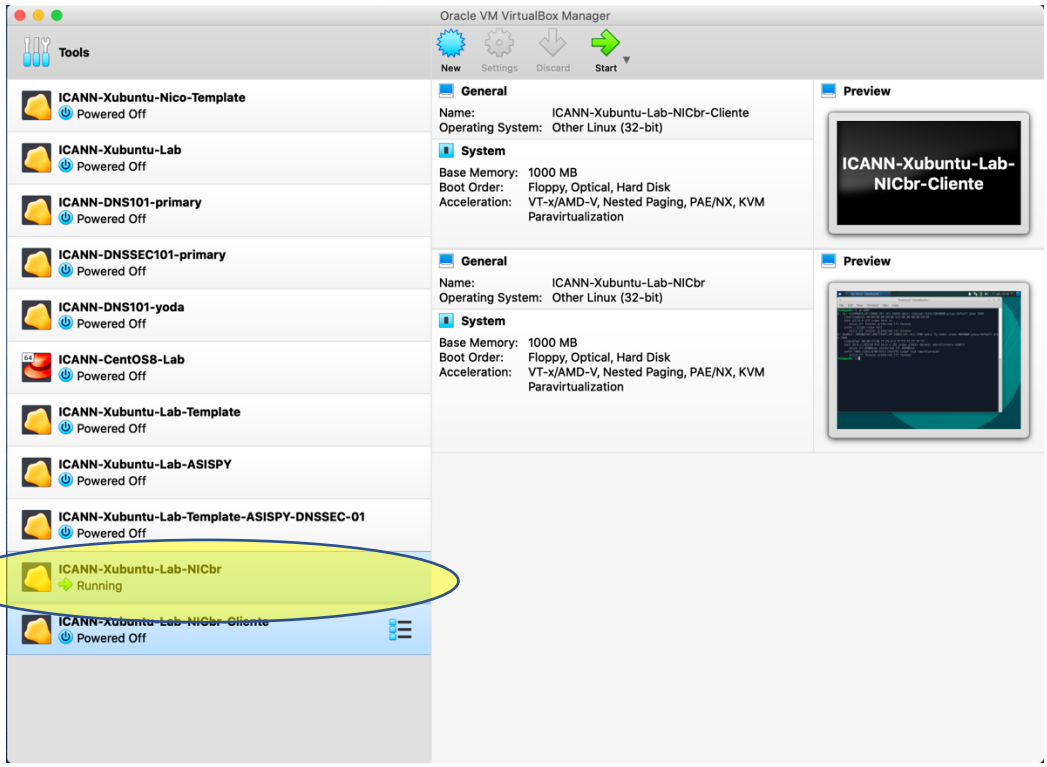
Hands-On Lab

Configuring BIND + DNSSEC + Hyperlocal

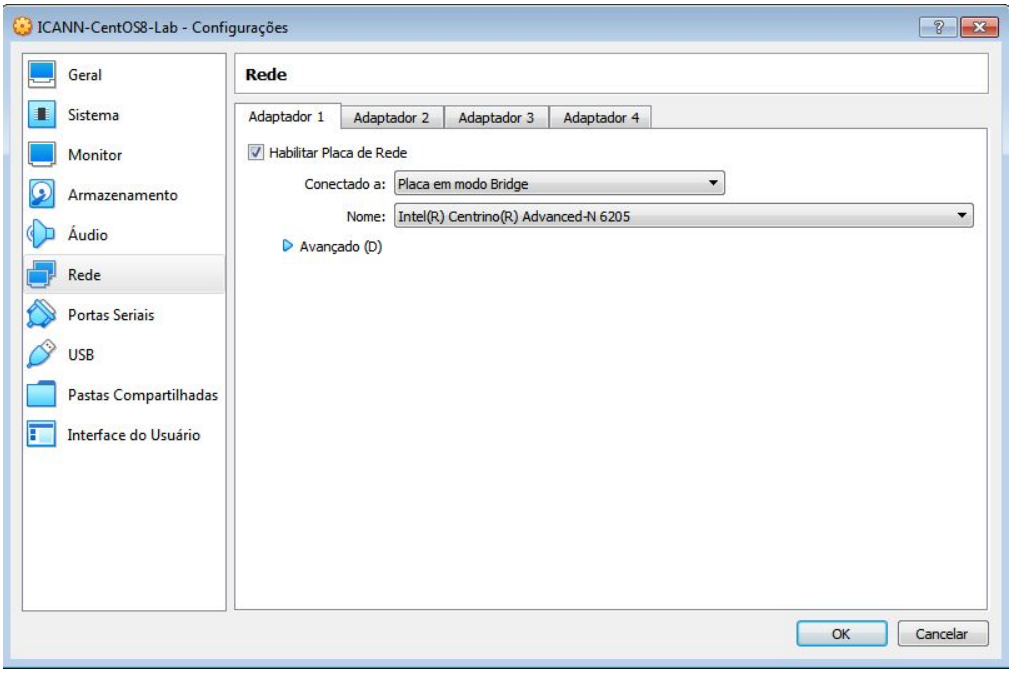
1) Preparing the System

a. Install Oracle Virtual Box

b. Import Xubuntu VM Image

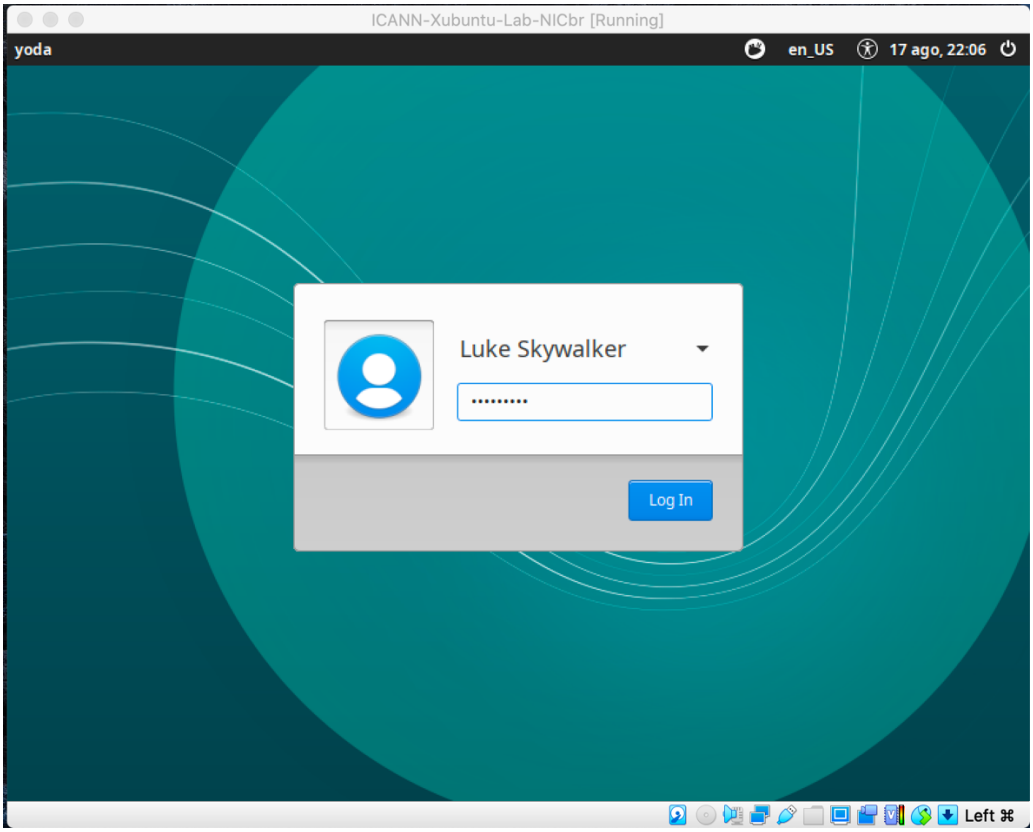


c. Configure VM network adaptor to operate in Bridge Mode



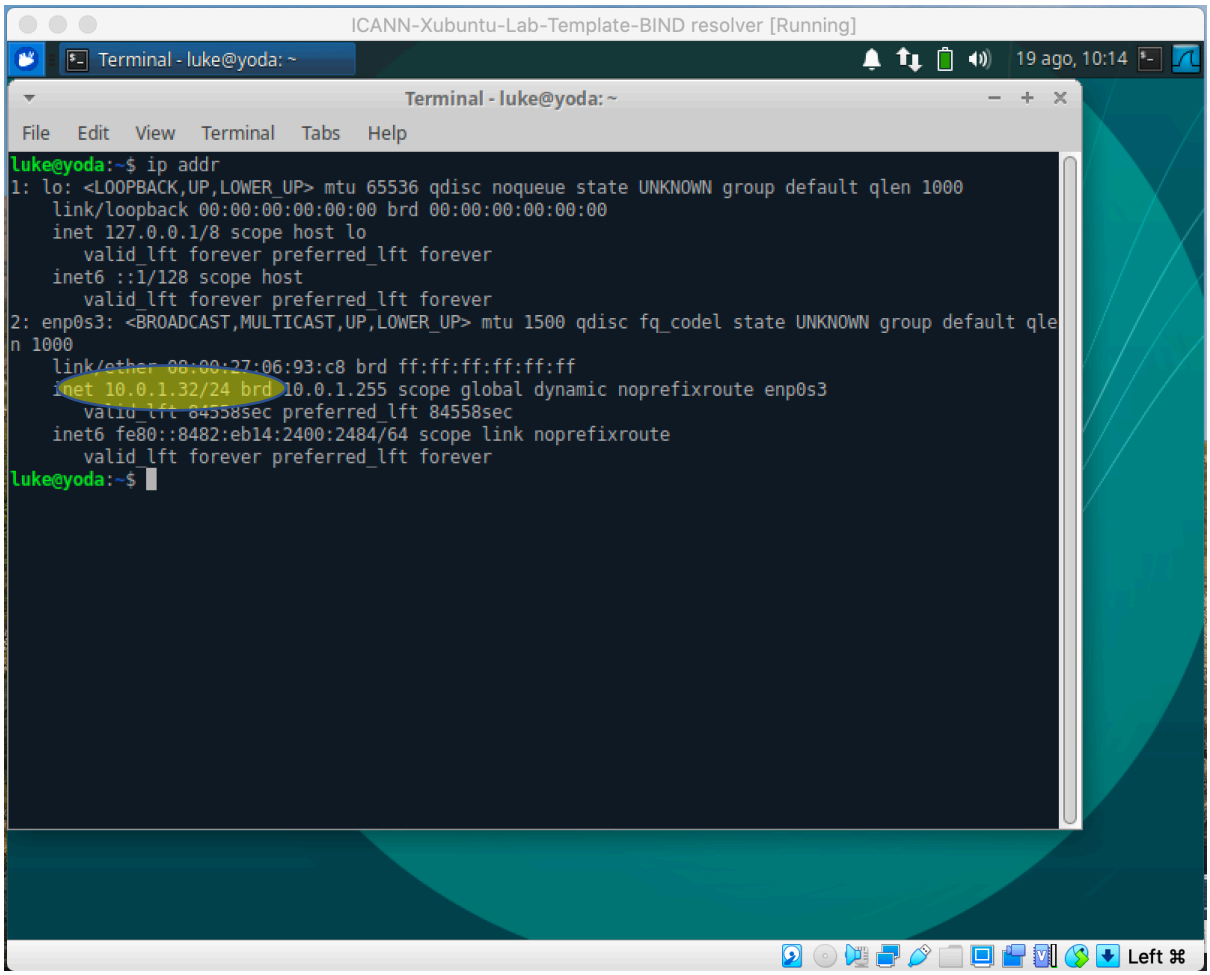
d. Start the VM

```
User:     Luke Skywalker
Pwd:      icannlab1
```



e. Find the VM IP address (VM name: “yoda”)

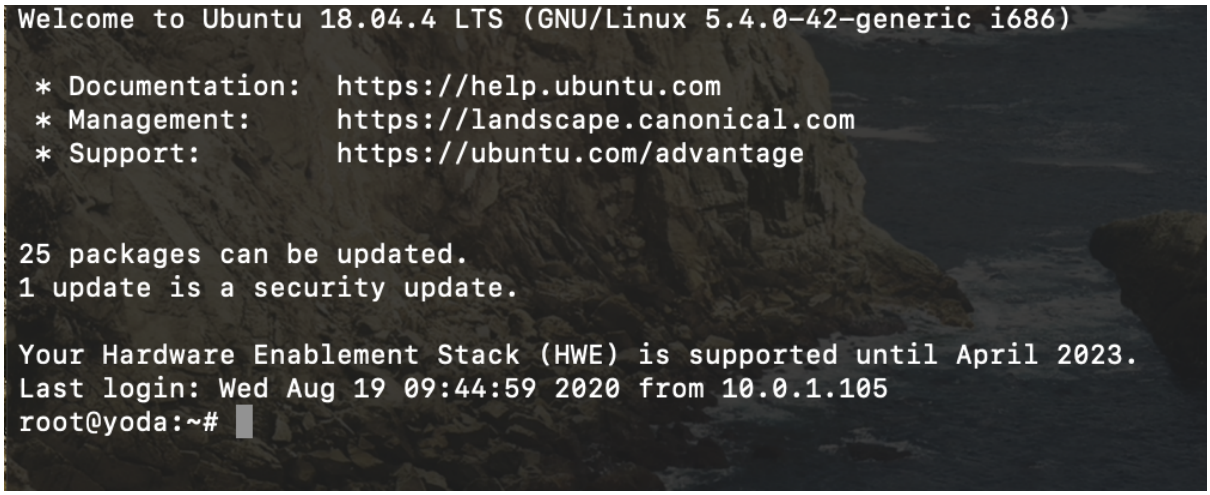
```
# ip addr

10.0.1.32       (Network: 10.0.1.0/24)
```



f. Connecting to the server to install and configure BIND

Now, in the configuration part, we will use an SSH client to connect to the server (yoda) with administrator privileges (root). To do this, if our machine has Linus or OSX, just open a terminal and connect to our server via SSH:

```
$ ssh root@10.0.1.32
```



If you are using Windows, you may use the SSH client application called PUTTY that (installed earlier) to connect to our server (yoda).

2) Setting up BIND server

a. Installing all necessary packages

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

To install the packages needed to run BIND, we will use the above commands (`apt-get`), one at a time.
If the installation process asks us if we want to install any of the packages, we will say `yes`.

The above commands install BIND packages, BIND documentation, BIND utilities (for example, to verify configuration, etc.) and DNS utilities.

b. It is convenient to add a rule to allow port 53 in case we have firewall running (Ubuntu often comes with UFW firewall running):

```
# ufw allow 53
```

3) Configuring BIND

a. Acces BIND folder
```
# cd /etc/bind
```

b. The standard installation will already bring the information for locating root servers
```
# more db.root
```

```
;       This file holds the information on root name servers needed
;       to initialize cache of Internet domain name servers
;       (e.g. reference this file in the "cache  .  <file>"
;       configuration file of BIND domain name servers).
;
;       This file is made available by InterNIC
;       under anonymous FTP as
;           file                /domain/named.cache
;           on server           FTP.INTERNIC.NET
;       -OR-                    RS.INTERNIC.NET
;
;       last update:    February 17, 2016
;       related version of root zone:   2016021701
;
; formerly NS.INTERNIC.NET
;
.                        3600000      NS    A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET.      3600000      A     198.41.0.4
A.ROOT-SERVERS.NET.      3600000      AAAA  2001:503:ba3e::2:30
;
; FORMERLY NS1.ISI.EDU
;
…

; OPERATED BY ICANN
;
.                        3600000      NS    L.ROOT-SERVERS.NET.
L.ROOT-SERVERS.NET.      3600000      A     199.7.83.42
L.ROOT-SERVERS.NET.      3600000      AAAA  2001:500:3::42
;
; OPERATED BY WIDE
;
.                        3600000      NS    M.ROOT-SERVERS.NET.
M.ROOT-SERVERS.NET.      3600000      A     202.12.27.33
M.ROOT-SERVERS.NET.      3600000      AAAA  2001:dc3::35
; End of file
```

4) Editing BIND configuration files

a. The configuration file (`named.conf.options`) for the standard installation is located in the BIND directory (`/etc/bind`)

```
# more named.conf.options
```

```
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 placeholder.

     // forwarders {
     //   0.0.0.0;
     // };

     //========================================================
     // 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;

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

b. We will perform the following actions in the configuration (explanation of the options)

- To simplify the practice, we have disabled the ability to receive queries using the IPv6 protocol (only queries using IPv4). However, we recommend that any recursive DNS server have the IPv4 and IPv6 protocols configured in all cases of production deployments.

  ```
  //listen-on-v6 { any; };
  ```

- For this first part of the practice, we will explicitly disable DNSSEC. Note that, by default, BIND will have DNSSEC validation enabled.

  ```
  dnssec-enable no;
  //dnssec-validation auto;
  ```

When dnssec-validation is set to automatic, the default is the DNS root zone as the trust anchor. BIND includes a copy of the root key that is kept up to date automatically. If set to yes, a trust anchor must be set up explicitly using the *managed-keys* or *trusted-keys* option.

- We explicitly enable recursion.

  ```
  recursion yes;
  ```

- We create an access list to allow only DNS queries from the server itself or from our network (to prevent the recursive server from being open to the world).

  ```
  listen-on port 53 { localhost; 10.0.1.0/24; };
  allow-query { localhost; 10.0.1.0/24; };
  ```

For the configuration file to be as follows:

```
# nano named.conf.options
```

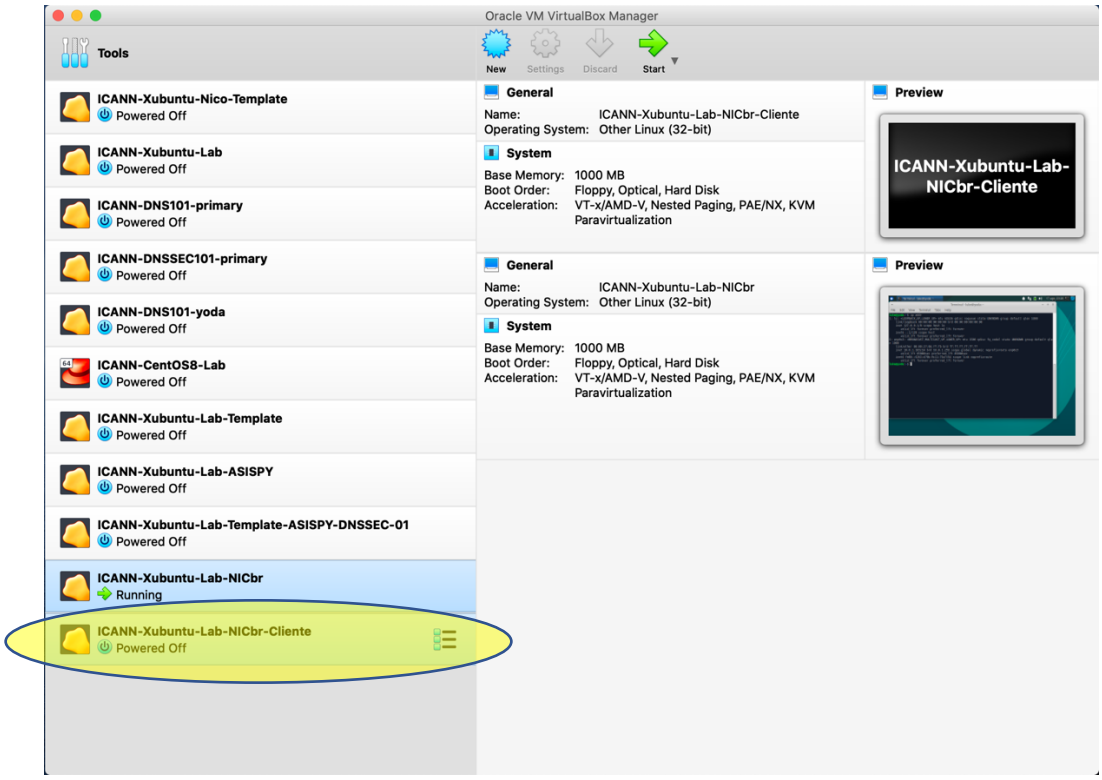
```
options {
     directory "/var/cache/bind";

     // 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 placeholder.

     // forwarders {
     //   0.0.0.0;
     // };

     //========================================================================
     // 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-enable no;
     //dnssec-validation auto;

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

     listen-on port 53 { localhost; 10.0.1.0/24; };
     allow-query { localhost; 10.0.1.0/24; };

     recursion yes;
};
```

c. Finally, we use the BIND functionality to confirm that there are no errors in the configuration files and restart the BIND server to apply the configuration changes

```
# named-checkconf
# service bind9 restart
```

5) Testing our BIND server using a client VM

Now, for the test part of our recursive server, we will use a new virtual machine. To do this, we run the second virtual machine (client) that we previously installed.

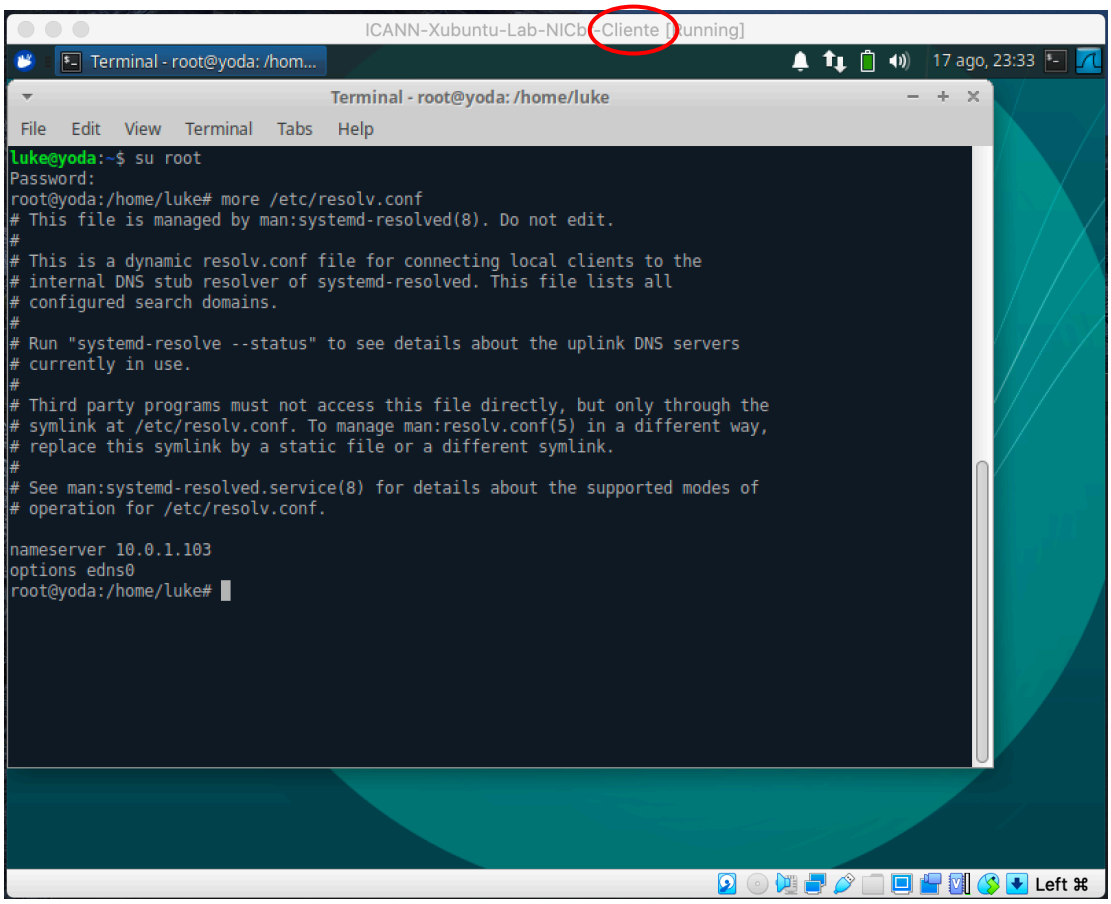


6) We may adjust the client configuration (in the *resolv.conf* file) to use our BIND server

   a. Open */etc/resolv.conf* and include the server address (using administrator user)

Remember to make this configuration on the CLIENT and not on the server !!!

```
$ su root       (Password: icannlab1)

# nano /etc/resolv.conf

nameserver 10.0.1.103
```



7) Using DIG command

 # dig [server] [domain-name] [requested RR] [options]

a. Just type *dig* into the terminal

```
$ dig
```

- What happens?
- What was the response time (query time)?

b. Make some tests using *dig* command

```
$ dig [your-favorite-domain-name] a
$ dig www.google.com a
$ dig lacnic.net mx
$ dig nonexistentdomain.sometld
$ dig afrinic.net aaaa
```

Now send queries to another to resolver

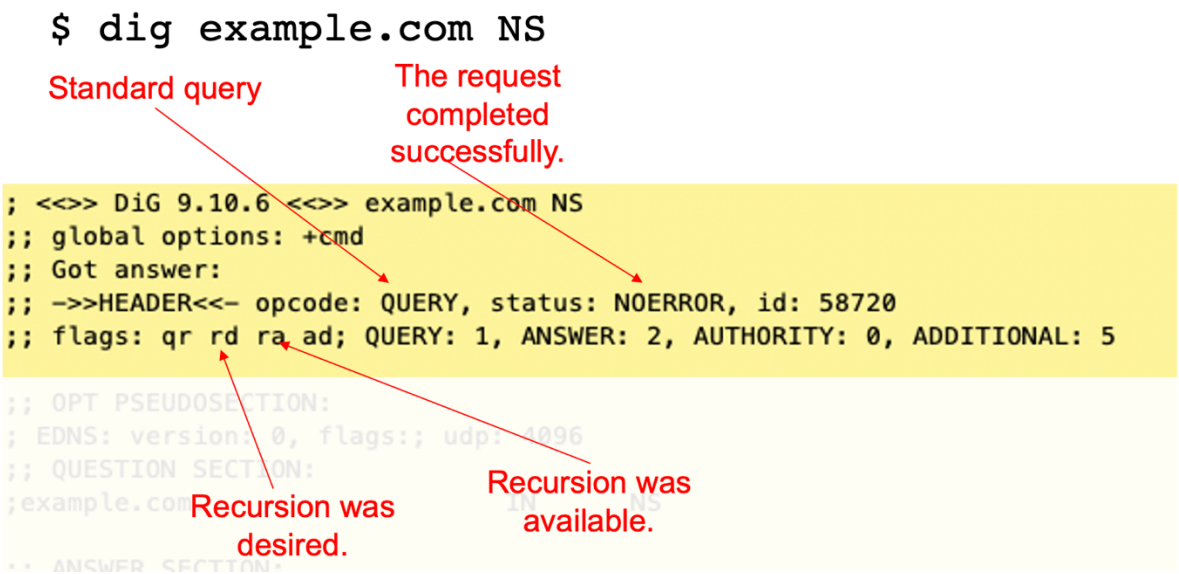
```
$ dig @8.8.8.8 o-seu-site-favorito a
$ dig @9.9.9.9 www.google.com a
```

Other tests. Compare response times.

Check the response time of the root servers

```
$ dig @l.root-servers.net id.server ch txt
$ dig @b.root-servers.net id.server ch txt
```

c. Recalling about *dig* response



Testing DNSSEC validation in our resolver

```
$ dig dnssec-failed.org
```

What happens?

8) Enabling **DNSSEC** validation

We will edit the configuration file to enable DNSSEC

```
# nano named.conf.options

dnssec-enable yes;
dnssec-validation auto;
```

We use the BIND functionality to confirm that there are no errors in the configuration files and restart the BIND server to apply the configuration changes

```
# named-checkconf          (If nothing is returned, everything is fine)
# service bind9 restart
```

a. Testing

```
$ dig nic.br +dnssec +multi
```

(observe the *ad* -authenticated data- flag)

```
$ dig dnssec-failed.org
```

(What is the status?)

b. Checking DNSSEC related Resource Records

DNSKEY
```
$ dig nic.br DNSKEY
```

DS
```
$ dig nic.br DS
```

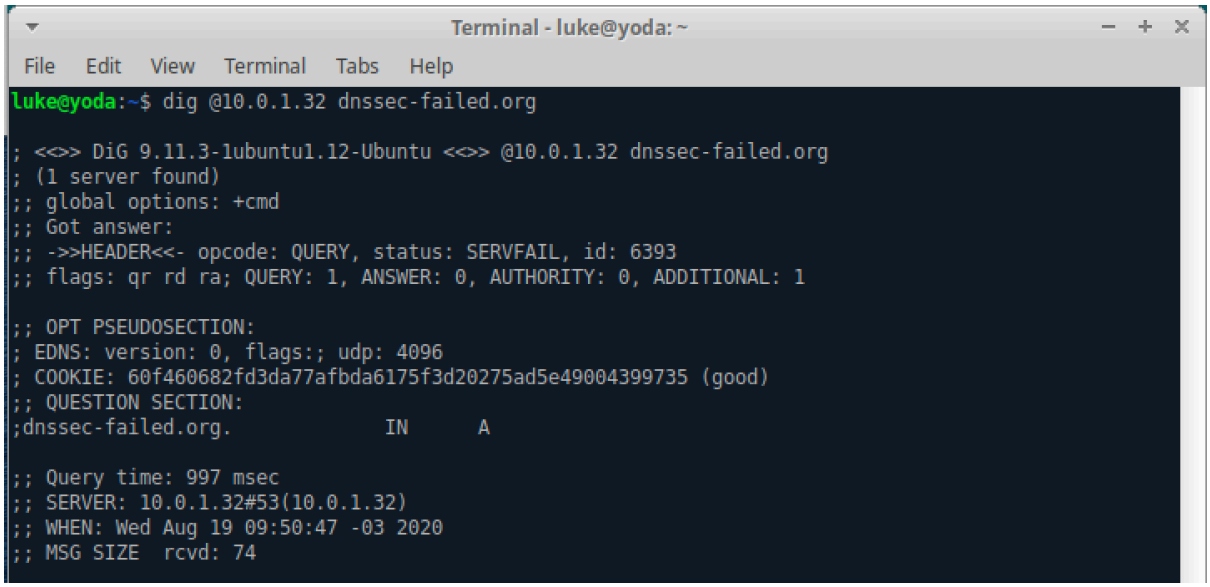
RRSIG
```
$ dig nic.br RRSIG
```

c. How to create an exception for DNSSEC in BIND

Negative trust anchors (NTAs) can be used to mitigate DNSSEC validation failures by disabling DNSSEC validation for specific domains.

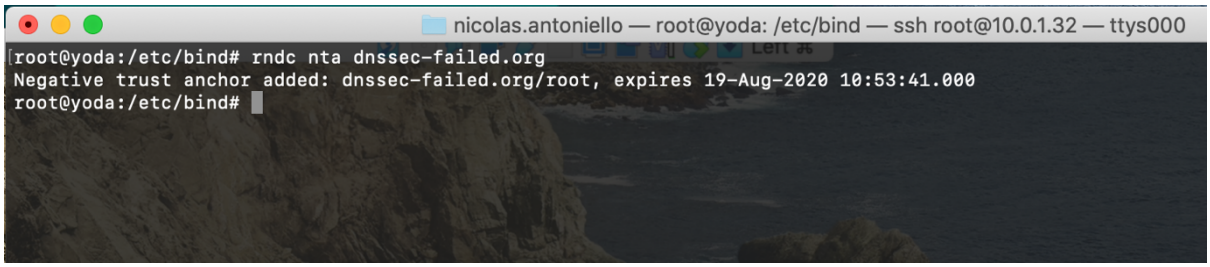
For example, we will use the domain *dnssec-failed.org* to test this feature.

First, from the client, we do a DNS query for the domain ***dnssec-failed.org***

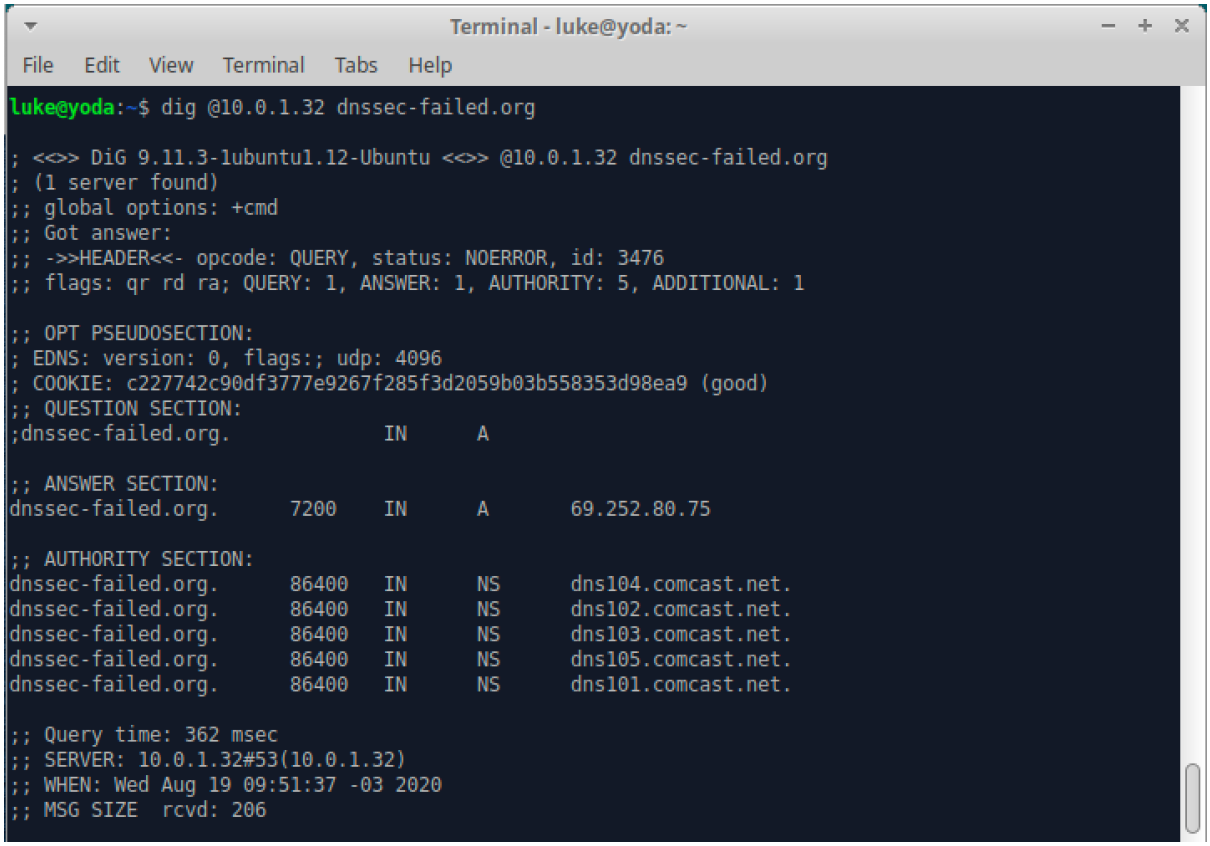


Now let's insert the exception on the DNS server using the command:

```
# rndc nta dnssec-failed.org
```



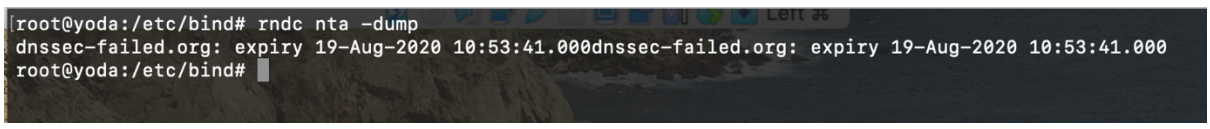
Finally, we do the DNS query again from the client for the same domain; now getting the answer without DNSSEC validation (because of the exception we made).



The default lifetime of an NTA is one hour, although by default, BIND polls the zone every five minutes to verify that the zone is now validated correctly, when the NTA will automatically expire. The default lifetime and polling interval can be configured via *named.conf*, and the lifetime can be changed by zone using the `-lifetime duration` parameter in *rndc nta* command. Both timer values have a maximum allowed value of one week.

A list of all configured NTAs can be viewed using the command:

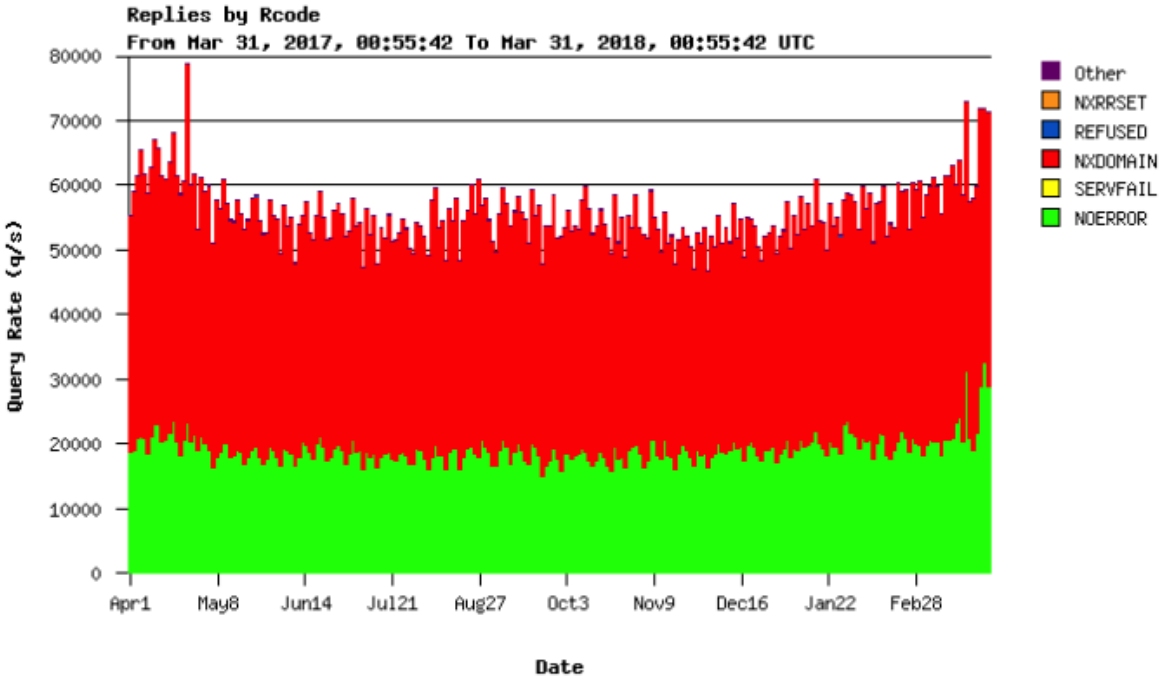
```
# rndc nta -dump
```



9) Setting up Hyperlocal on our DNS resolver

From RFC7706:
"*Some DNS recursive resolvers have longer-than-desired round-trip times to the closest DNS root server. Some DNS recursive resolver operators want to prevent snooping of requests sent to DNS root servers by third parties. Such resolvers can greatly decrease the round-trip time and prevent observation of requests by running a copy of the full root zone on a loopback address (such as 127.0.0.1). This document shows how to start and maintain such a copy of the root zone that does not pose a threat to other users of the DNS, at the cost of adding some operational fragility for the operator*"

Curiosity: 65% of the root server load is dedicated to NXDOMAIN



Yearly graph of return codes on K-Root, provided by the RIPE NCC

Source: https://www.ripe.net/analyse/dns/k-root/statistics/?type=ROOT&increment=yearly

To configure the hyperlocal, we will make some configuration changes to the *named.conf* and *named.conf.options* files. For that we will generate 2 views in BIND, one to answer the queries to the Root zone (the one that will keep the copy of that zone: hyperlocal) and the other to answer the recursive queries as a recursive server normally does. What will happen then is that the recursive server instead of consulting the root servers to resolve the "." it will consult by obtaining the information from the local database *rootzone.db* (as a result of the transfer of the root zone with the hyperlocal configuration).

a. Some test before configuring Hyperlocal

Recursive BIND without Hyperlocal:

Note that we will do a query test (we will query from the client and not the server) for a non-existent top-level domain (NXDOMAIN) to check the approximate response time when querying a root server.

```
$ dig domaininvalid.com.br.xxxxxxx
```

And we note the response time for this query.

b. We edit the *named.conf.options* configuration file as follows:

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

```
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 placeholder.

        // forwarders {
        //      8.8.8.8;
        // };

        //========================================================================
        // 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-enable yes;
        dnssec-validation auto;

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

        listen-on port 53 { localhost; 10.0.1.0/24; };
        allow-query { localhost; 10.0.1.0/24; };

        //recursion yes;  ← only commented this line to include it elsewhere
};
```

c. We edit the *named.conf* configuration file as follows:

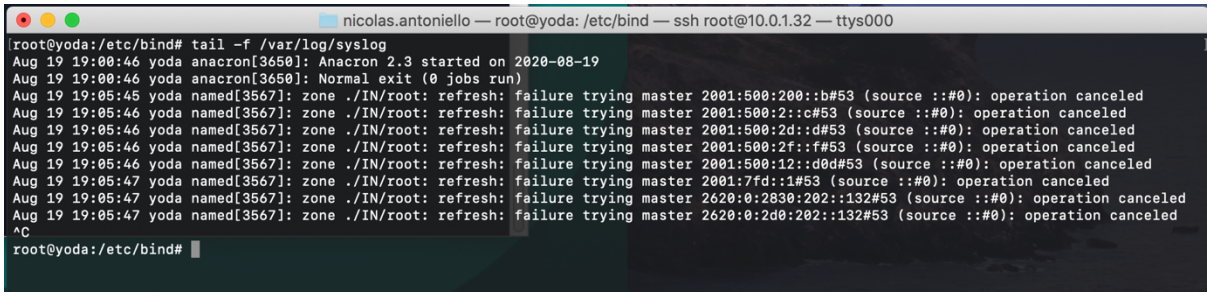
```
# nano /etc/bind/named.conf
```

```
// This is the primary configuration file for the BIND DNS server named.
//
// Please read /usr/share/doc/bind9/README.Debian.gz for information on the
// structure of BIND configuration files in Debian, *BEFORE* you customize
// this configuration file.
//
// If you are just adding zones, please do that in /etc/bind/named.conf.local

include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";

// We comment the following -include- because we will not use this configuration.
// Instead, we will use the transfer of the root zone (hypelocal).
//include "/etc/bind/named.conf.default-zones";

// We create the view for the root zone
view root {
        // We indicate which recursive servers can use the local copy of the root
        match-clients { localhost; };
        zone "." {
                type slave;
                file "rootzone.db"; // This file stores the local copy of the root
                notify no;
                masters {
                        199.9.14.201;           # b.root-servers.net
                        192.33.4.12;            # c.root-servers.net
                        199.7.91.13;            # d.root-servers.net
                        192.5.5.241;            # f.root-servers.net
                        192.112.36.4;           # g.root-servers.net
                        193.0.14.129;           # k.root-servers.net
                        192.0.47.132;           # xfr.cjr.dns.icann.org
                        192.0.32.132;           # xfr.lax.dns.icann.org
                        2001:500:200::b;        # b.root-servers.net
                        2001:500:2::c;          # c.root-servers.net
                        2001:500:2d::d;         # d.root-servers.net
                        2001:500:2f::f;         # f.root-servers.net
                        2001:500:12::d0d;       # g.root-servers.net
                        2001:7fd::1;            # k.root-servers.net
                        2620:0:2830:202::132;   # xfr.cjr.dns.icann.org
                        2620:0:2d0:202::132;    # xfr.lax.dns.icann.org
                };
        };
};

// We create the view for answering recursive queries
view recursive {
        dnssec-enable yes;
        dnssec-validation auto;

        allow-recursion { localhost; 10.0.1.0/24; };
        recursion yes;

        // To solve the root we will use the local root zone copy (hyperlocal)
        zone "." {
                type static-stub;
                server-addresses { 127.0.0.1; };
        };
};
```

d. We use the BIND functionality to confirm that there are no errors in the configuration files and restart the BIND server to apply the configuration changes

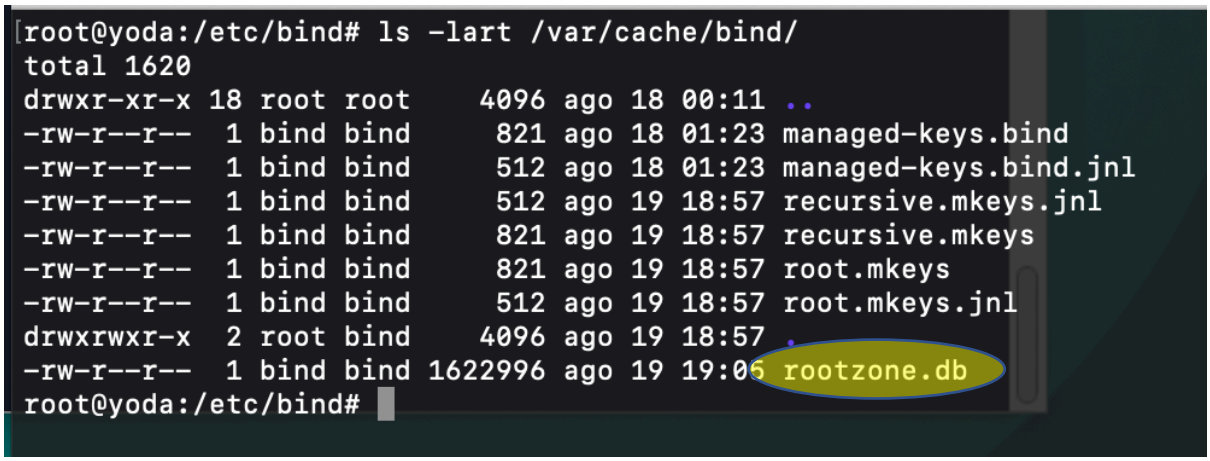
```
# named-checkconf     (No news, good news!)
# service bind9 restart
```

e. We can check the system log */var/log/syslog* to verify that BIND has been restarted correctly (we will certainly have warnings that IPv6 destinations are not accessible if we do not have IPv6 connectivity).



f. We check if the local file with the copy of the root zone (*rootzone.db*) was created correctly:

```
# ls -lart /var/cache/bind/
```



g. Testing after Hyperlocal configuration

Recursive BIND with Hyperlocal

We will perform some query tests (we will query from the client and not from the server) for nonexistent top-level domains (NXDOMAIN) to check the approximate response time when querying different root servers.

Test 1: Google public resolver

```
$ dig @8.8.8.8 nonexistingdom.com.xxxxxxx
```

Test 2: My ISP recursive server

```
$ dig @[my-gateway-ip-add] nonexistingdom.com.xxxxxxx
```

Test 3: Our recursive BIND server with Hyperlocal

```
$ dig @10.0.1.32 nonexistingdom.com.xxxxxxx
```

We compare the time obtained now with that obtained when we did not have Hyperlocal configured.

END.