LPIC-2 TRAINING COURSE

Topic 207: DNS & BIND

Domain Name System

- **Provides a mapping from names to resources of several types**
- **Naming history: using host.txt file in every single machine**
  - Traffic and load?
  - Name collisions?
  - Consistency?
- **DNS created in 1983, commprised of three components**
  - A domain or “name space”
  - Servers making that name space available
  - Resolvers (clients) with query the servers about the name space

DNS Features

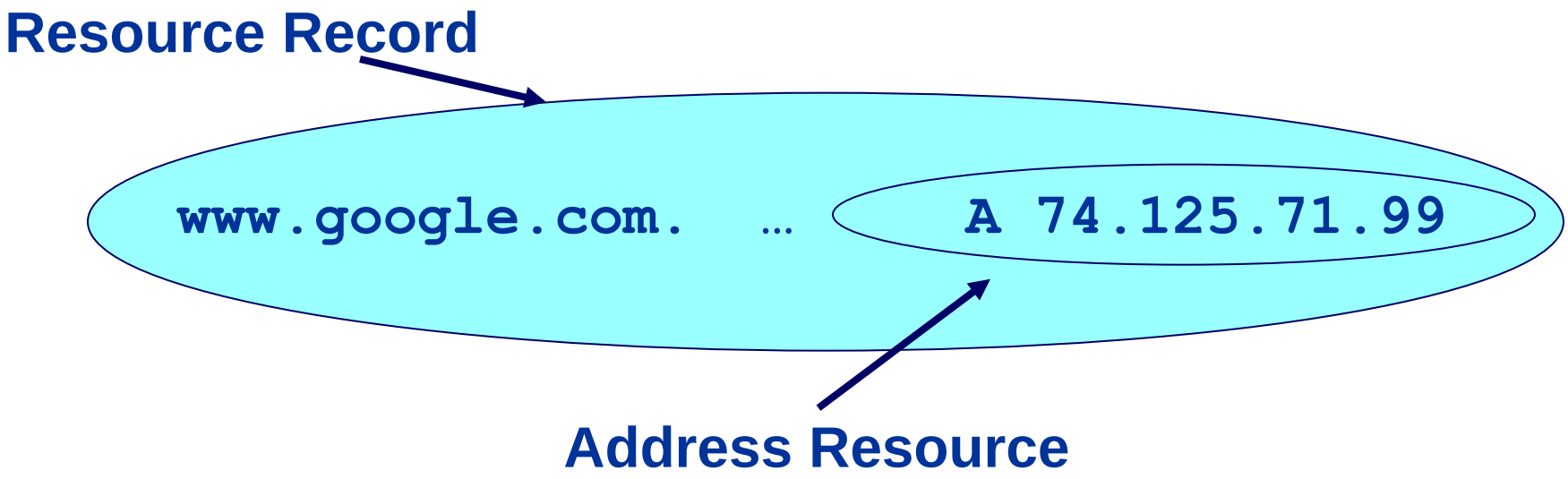
- **Global distribution: database is maintened locally, but retriveable globally**
  - No single computer has all DNS data
- **Loose coherency: database is always internally consistent**
  - Cached data expires according to timeout value
- **Scalability: no limit to the size of the database or to the number of queries**
- **Reliability: clients can query master or slave servers**
  - Database is replicated from master to multiple slaves
- **Dynamicity: database can be updated dynamically**
  - Modification of the master database triggers replication

DNS Concepts: FQDN and RR

- **Fully Qualified Domain Name (FQND)**

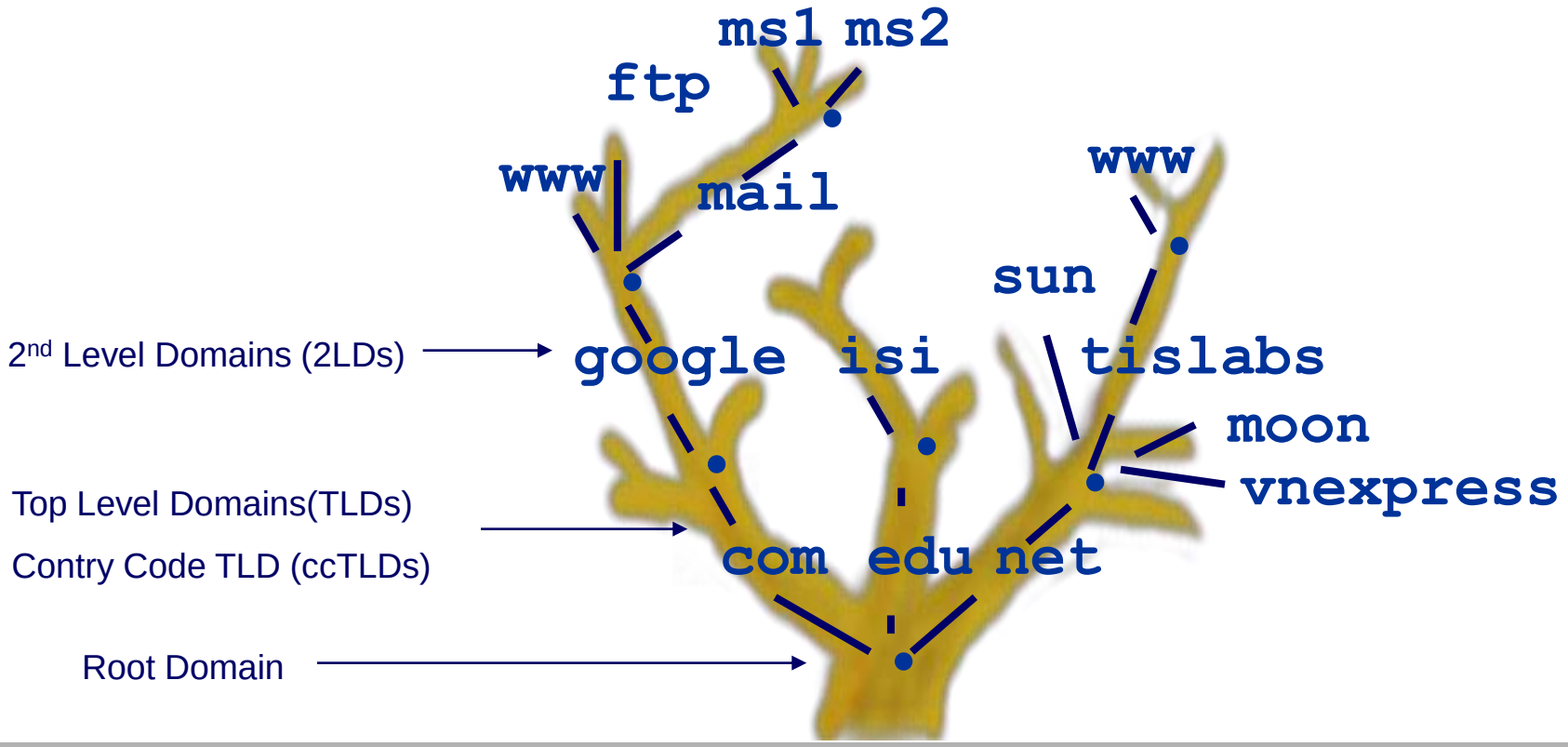
  www.google.com. ← *Note the trailing dot*

  - Labels seperated by dots.
- **DNS maps names into data using Resource Record (RR)**



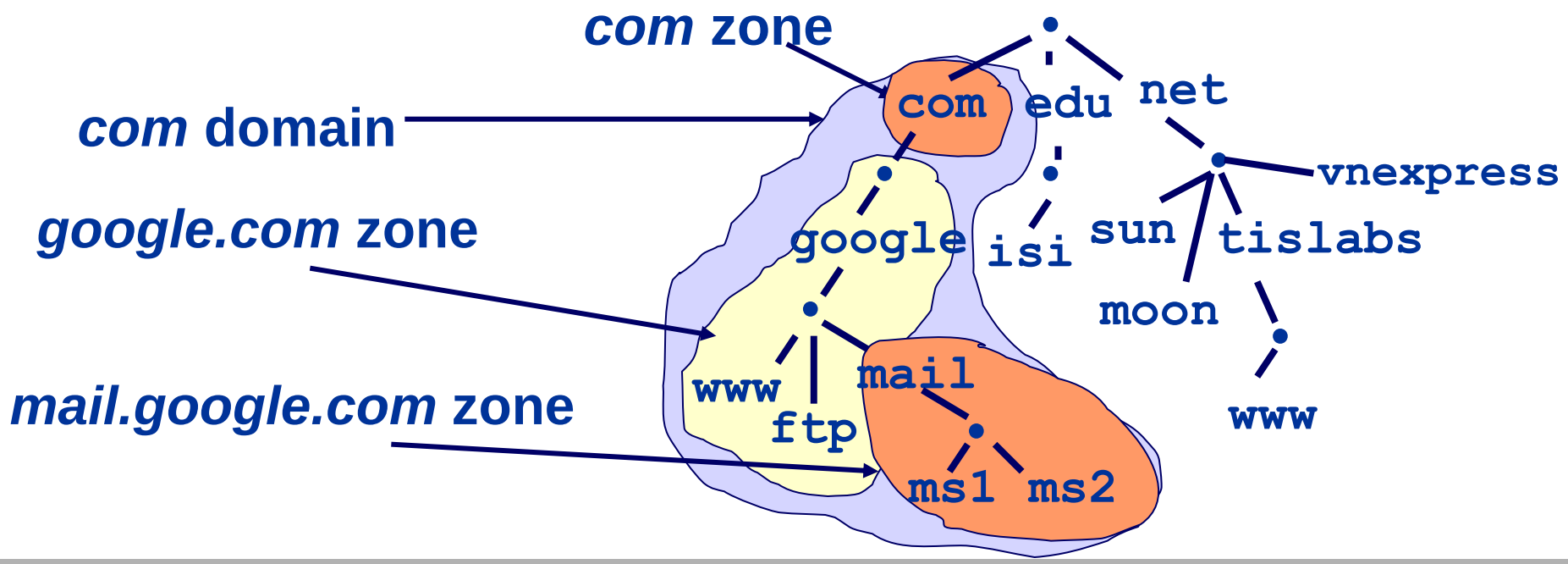
DNS Concepts: Domain

- Domain are “namespaces”, can be mapped to a tree
  - . is the “*root*” domain
  - Everything below .**net** is in the *net* domain
  - Eveything below **google.com** is in the *google.com* domain and in the *net* domain



DNS Concepts: Zone & Delegation

- Administrators of a domain can create subdomains and delegate responsibility for managing subdomain to someone
- Zone are “administrative spaces”
  - Zone administrators are responsible for portion of a domain’s name space
  - Zone refers to all the resource record in a domain, not in its sub domain



DNS Concepts: Name Servers

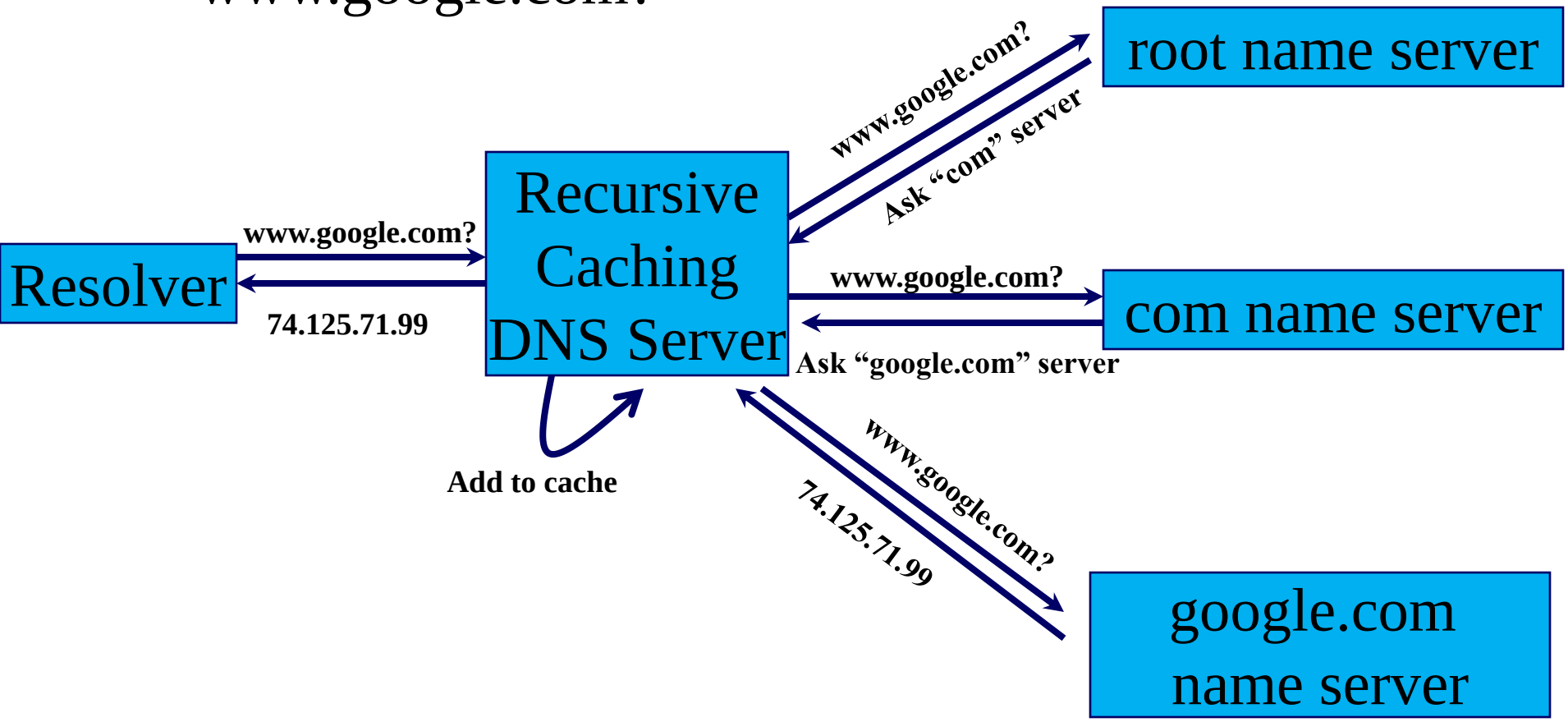
- **Name servers answer "DNS" questions**
- **Types of name servers:**
  - Authoritative servers: give authoritative answers for one or more zones
    - Master (primary): loads the data from a zone file
    - Slave (secondary): replicates the data from the master via a zone transfer
  - (Caching) recursive servers: obtains answer from authorative server and forward to the clients
    - Answer are marked as not authorative
    - Answer are stored for future reference in the cache
  - Mixture of functionality

DNS Concepts: Resolvers

- **Client-side of the DNS**
- **Send query to the DNS server on behalf of the application**
- **Two type of DNS query:**
  - **Non-recursive query**
    - DNS server provide a authoritative record
    - Or provide partial result without querying other servers
  - **Recursive query**
    - DNS server provide a authoritative record
    - DNS server provide a fully answer by querying other name server as needed

Example: Resolving Process

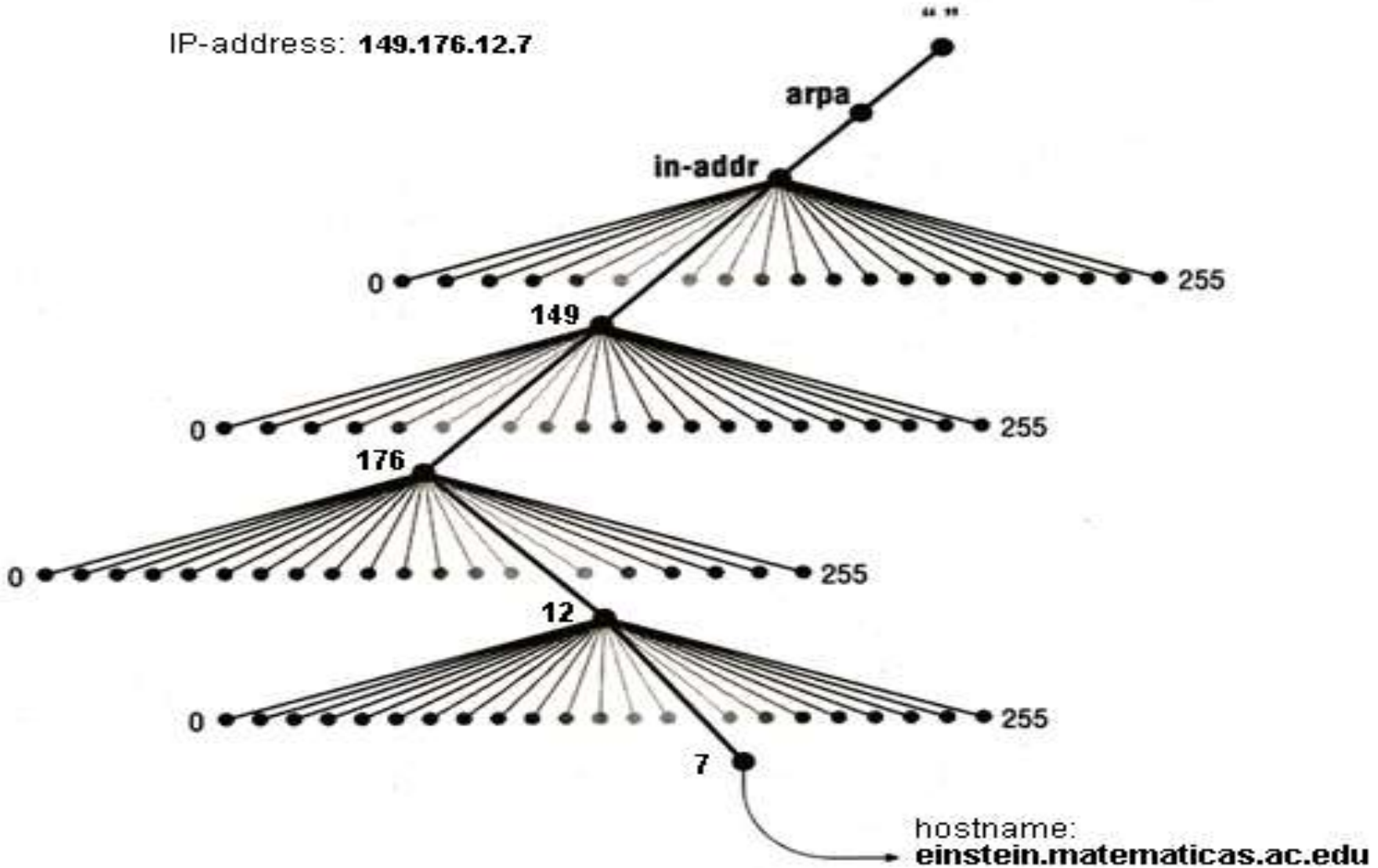
Question:
www.google.com?



Reverse Lookup

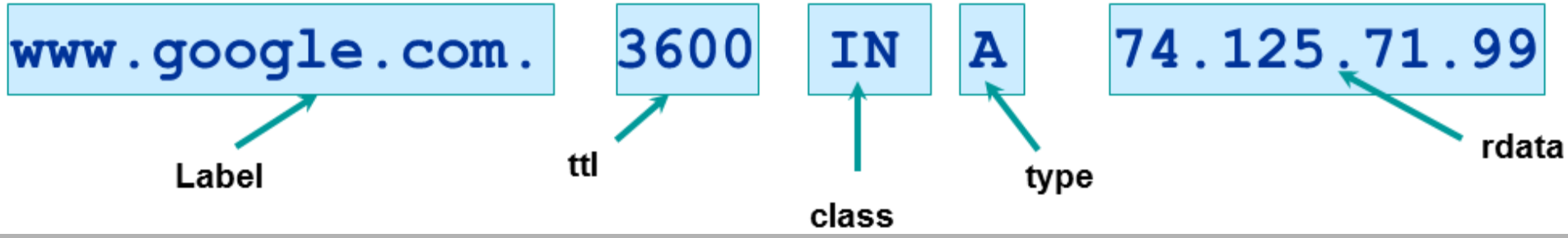
- **Query of the DNS for domain names when the IP address is known**
- **`in-addr.arpa` is the reverse lookup domain for IPv4**
- **IP address is represented as a name in reverse-ordered octet**
  - Eg: IPv4 address **`149.176.12.7`** is know as a DNS name of: **`7.12.176.149.in-addr.arpa`**
- **Every zone file must have a *reverse zone* file for each IP range**

In-addr.arpa Domain



Resource Records (RR)

- Normally defines in “zone files”
- Consist of name, TTL, Class, Type and RDATA
  - **TTL**: Time To Live paramater
  - **Class**: **`IN`** is widest used
  - **Type**: **`SOA, NS, A, CNAME, MX, PTR`**…
  - **RDATA**: everything behind the **Type** identifier



RR Example: Zone File

```
ipmac.vn.              7200 IN SOA   ns.ipmac.vn. admin.ipmac.vn. (
                                2002021301    ; Serial
                                24h           ; Refresh
                                2h            ; Retry
                                30d           ; Expiration
                                1h )          ; Negative caching TTL

ipmac.vn.              7200 IN NS          ns.ipmac.vn.
in.ipmac.vn.           7200 IN NS          in-ns.ipmac.vn.

ns.ipmac.vn.           3600 IN A           115.84.117.1
in-ns.ipmac.vn.        3600 IN A           192.168.0.1
www                    3600 IN A           115.84.117.59
mail1.ipmac.vn.        3600 IN A           115.84.117.60
mail2                  3600 IN A           115.84.117.61

web                    3600 IN CNAME       www

ipmac.vn.              5400 IN MX          0      mail1
ipmac.vn.              3600 IN MX          10     mail2.ipmac.vn.
```

RR Example: Reverse Zone File

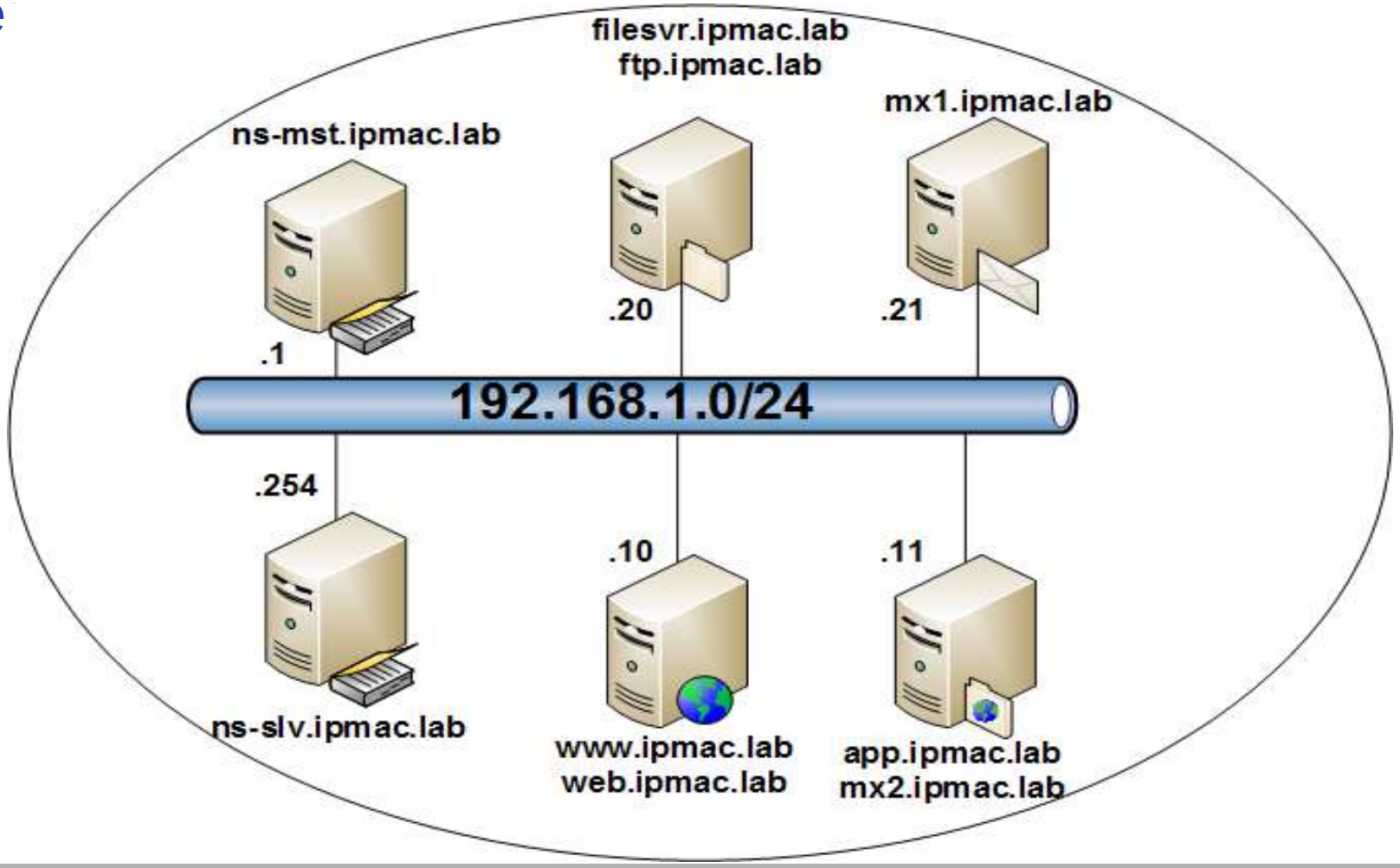
```
117.84.115.in-addr.arpa. 7200 IN SOA  ns.ipmac.vn.  admin.ipmac.vn. (
                              2002021301     ; Serial
                              24h            ; Refresh
                              24h            ; Retry
                              30d            ; Expiration
                              1h )           ; Neg. TTL

117.84.115.in-addr.arpa.      7200 IN NS            ns.ipmac.vn.

1.117.84.115.in-addr.arpa.    3600 IN PTR           ns.ipmac.vn.
59.117.84.115.in-addr.arpa.   3600 IN PTR           www.ipmac.vn.
60                            3600 IN PTR           mail1.ipmac.vn.
61                            3600 IN PTR           mail2.ipmac.vn.
```

Exercise 1: Creating Zone & Reverse Zone file

Create the zone file named **ipmac.lab.zone** and the reverse zone file named **192.168.1.rv.zone** for the following administrative zone



Berkeley Internet Name Domain - BIND

- **Most commonly used DNS server**
  - Newest version is BIND 9
- **BIND 9 components:**
  - Daemon: **`named`**
  - Configuration file: **`/etc/named.conf`**
  - Default working directory: **`/var/named/`**
    - Zone files are normally placed in the working directory
  - Daemon control program: **`rncd`**

BIND Configuration File (named.conf)

- **Contain *statements* that start with a `keyword` plus a { and end with a }**
- **Comments are started with // or #**
- **Some *statements*:**
  - `options { … }` : **named** options
  - `logging { … }` : log options
  - `zone “zonename” { … }` : zone definition
- **Example:**

```
zone “example.org" {
        type master; //master of this zone
        file “/var/bind/example.org.zone";
};
```

BIND Zone Files

Example of `example.org.zone` file

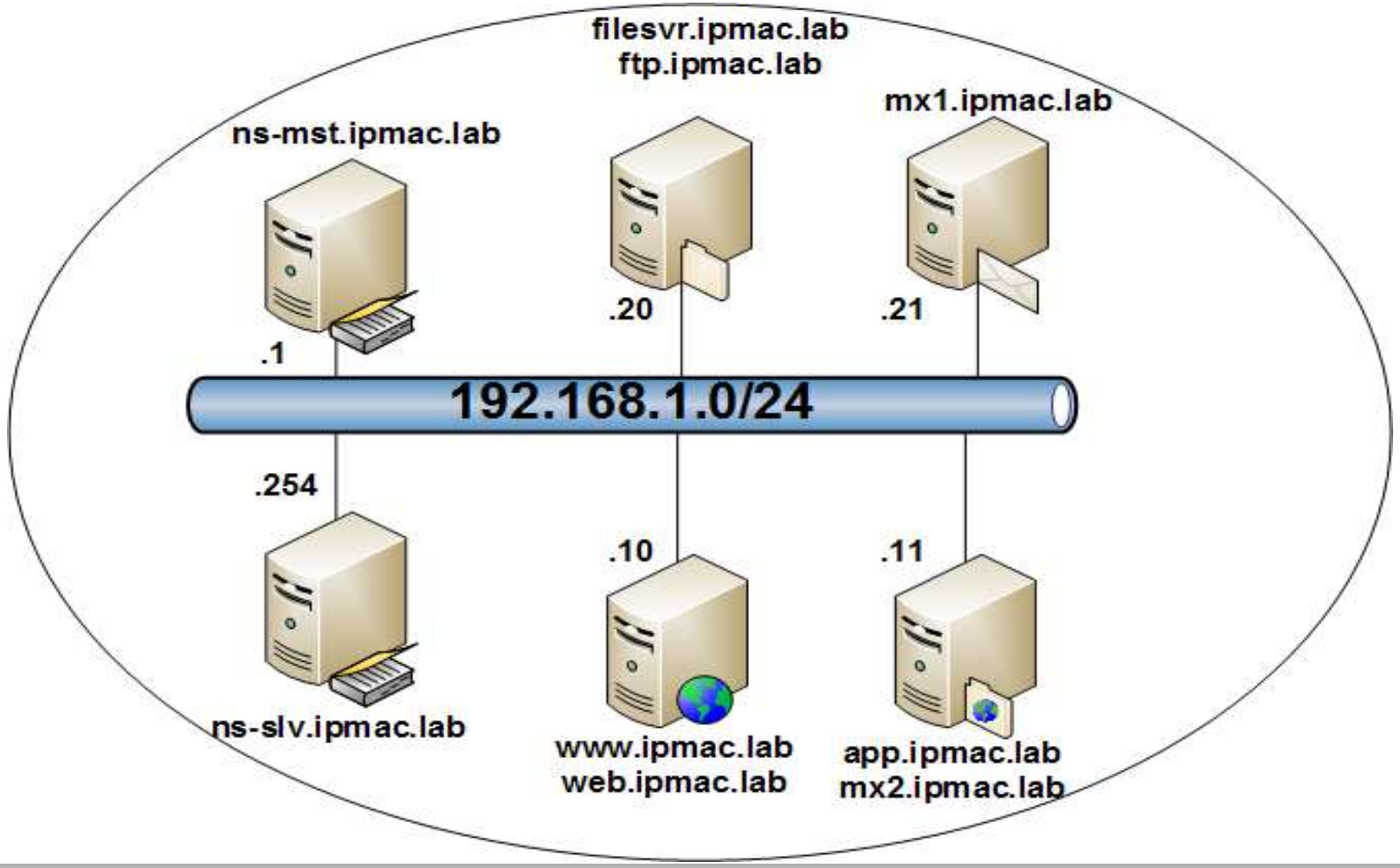
```
$TTL 86400
@   IN     SOA ns.example.org  admin.example.org (
                    … )
    IN     NS     ns.example.org.
    IN     NS     ns1.example.org.
web IN     A      192.168.0.1
…
```

Explain:

- **`$TTL:`** Default Time To Live for the zone
- **`@:`** Current origin, which expands to **example.org** according to **named.conf** file. Current orgin will be added to any name that does not end in a dot
- TTL in each RR may be ommited (use default TTL)

Exercise 2: Creating Zone & Reverse Zone file for BIND

Change the zone & reverse zone file you created in Exercise 1 to BIND-compatible zone file:



Manipulating DNS with `dig`

- Query for the A record of example.org:
  ```
  $ dig example.org A
  ```
- Query another name server for the MX record of example.org:
  ```
  $ dig @ns1.dns.net example.org MX
  ```
- Query the reverse zone definition:
  ```
  $ dig 1.131.113.203.in-addr.arpa PTR
  ```
- Query the version from a running name server:
  ```
  $ dig @ns1.dns.net chaos version.bind txt
  ```
- Get the complete information about example.org from a name server:
  ```
  $ dig @ns1.dns.net example.org axfr
  ```
- Download root zone file from root server:
  ```
  $ dig @a.root-servers.net . ns > root.db
  ```

Securing BIND

1. **Have redundant servers**
2. **Hiding your BIND version number**

```
options { …; version “hidden”; …; };
```

3. **Allow queries only from dedicated servers**

```
zone … { …; allow-queries {192.168.1.0/24;}; …; };
```

4. **Restrict zone transfers**

```
zone … { …; allow-transfer {10.10.10.5;192.168.1.12;}; …; };
```

5. **Encrypt zone transfers between DNS servers using Transaction Signatures (TSIG)**
6. **Do not allow recursive queries**

```
options { …; recursion no; …; };
```

7. **Recursive queries should be allowed from internal network only**

```
zone … { …; allow-recursion {192.168.1.0/24; 10.10.0.0/16;};};
```

8. **Run BIND in a *chroot* jail and/or as a user with the minimum privileges**

```
# named -u named -g named -t /chroot
```

   - RHEL/CentOS: using **`bind-chroot`** package

9. **Keep backups of your zone files in a separate environment**

Exercise 3: Install and Configure BIND

In this exercise, you will install ***bind*** and ***bind-chroot***, then configure BIND to run in *chroot* environment and provide DNS service for the authorative zone ***ipmac.lab*** with the zone files you created in Exercise 2.

1. Verify that **bind**, **bind-chroot** , **bind-utils** and **system-config-bind** packages are installed. If not, install these packages before continue
   Hint: `# rpm -q bind bind-chroot bind-utils system-config-bind`
2. Search for the BIND configuration template and zone files provided by **system-config-bind** package
   Hint: `# find / -name named* | grep system-config-bind`
3. Copy BIND configuration template and zone files to the proper directories under ***/var/named/chroot***
   Hint:
   ```
   # cd /usr/share/system-config-bind/profiles/default/
   # cp ./named.conf /var/named/chroot/etc/
   # cp ./named/* /var/named/chroot/var/named
   ```
4. Download the **named.root** file from ftp.rs.internic.net to ***/var/named/chroot/var/named***
   Hint:
   ```
   # cd /var/named/chroot/var/named
   # wget ftp://ftp.rs.internic.net/domain/named.root
   ```
5. Start **named** and verify that it's runing in chroot jail, with less privileges account
   Hint: `# service named start && ps -ef | grep named`
6. At this moment, you have **named** running in Cache-Only mode. Configure your client to use this DNS server and verify that you can resolve any available hostname on Internet

Exercise 3: Install and Configure BIND (cont')

7. Add the zone definition statement for **ipmac.lab** zone to **named.conf**

   Hint: **`# vi /var/named/chroot/etc/named.conf`**

```
…
zone "ipmac.lab" IN {
        type master;
        file "ipmac.lab.zone";
};
zone "1.168.192.in-addr.arpa" IN {
        type master;
        file "192.168.1.rv.zone"
};
```

8. Force **named** to load the new configuration

   Hint: **`# service named reload`**

9. From your client, verify that you can resolve all configured hostnames in **ipmac.lab** zone

   Hint: **`# nslookup web.ipmac.lab`**

   `# nslookup 192.168.1.11`

Thank You !

BACKUP SLIDES

Exercise 1: ipmac.lab.zone

```
ipmac.lab.    86400 IN SOA  ns-mst.ipmac.lab admin.ipmac.lab. (
                            2002021301    ; Serial
                            24h           ; Refresh
                            2h            ; Retry
                            30d           ; Expiration
                            1h )          ; Negative caching TTL
ipmac.lab.    7200  IN NS         ns-mst.ipmac.lab.
ipmac.lab.    7200  IN NS         ns-slv.ipmac.lab.

ns-mst        3600  IN A          192.168.1.1
ns-slv        3600  IN A          192.168.1.254
www           3600  IN A          192.168.1.10
filesvr       3600  IN A          192.168.1.20
app           3600  IN A          192.168.1.11
mx1           3600  IN A          192.168.1.21
web           3600  IN CNAME      www
ftp           3600  IN CNAME      filesvr

ipmac.vn.     5400  IN MX         0      mx1
ipmac.vn.     5400  IN MX         10     mx2
```

Exercise 1: 192.168.1.rv.zone

```
1.168.192.in-addr.arpa.  86400 IN SOA  ns-mst.ipmac.lab admin.ipmac.lab.(
                               2002021301     ; Serial
                               24h            ; Refresh
                               2h             ; Retry
                               30d            ; Expiration
                               1h )           ; Negative caching TTL

1.168.192.in-addr.arpa. 7200 IN NS          ns-mst.ipmac.lab.
1.168.192.in-addr.arpa. 7200 IN NS          ns-slv.ipmac.lab.

1               3600 IN PTR             ns-mst.ipmac.lab.
254             3600 IN PTR             ns-slv.ipmac.lab.
10              3600 IN PTR             www.ipmac.lab.
20              3600 IN PTR             filesvr.ipmac.lab.
11              3600 IN PTR             app.ipmac.lab.
21              3600 IN PTR             mx1.ipmac.lab.
```

Exercise 2: ipmac.lab.zone

```
$TTL 1D
@               IN SOA  ns-mst.ipmac.lab admin.ipmac.lab. (
                                2002021301    ; Serial
                                24h           ; Refresh
                                2h            ; Retry
                                30d           ; Expiration
                                1h )          ; Negative caching TTL
                IN NS           ns-mst.ipmac.lab.
                IN NS           ns-slv.ipmac.lab.

ns-mst          IN A            192.168.1.1
ns-slv          IN A            192.168.1.254
www             IN A            192.168.1.10
filesvr         IN A            192.168.1.20
app             IN A            192.168.1.11
mx1             IN A            192.168.1.21
web             IN CNAME        www
ftp             IN CNAME        filesvr
ipmac.vn.       IN MX           0       mx1
ipmac.vn.       IN MX           10      mx2
```

Exercise 2: 192.168.1.rv.zone

```
$TTL 1D
@               IN SOA  ns-mst.ipmac.lab admin.ipmac.lab.(
                                2002021301   ; Serial
                                24h          ; Refresh
                                2h           ; Retry
                                30d          ; Expiration
                                1h )         ; Negative caching TTL

                IN NS           ns-mst.ipmac.lab.
                IN NS           ns-slv.ipmac.lab.

1               IN PTR          ns-mst.ipmac.lab.
254             IN PTR          ns-slv.ipmac.lab.
10              IN PTR          www.ipmac.lab.
20              IN PTR          filesvr.ipmac.lab.
11              IN PTR          app.ipmac.lab.
21              IN PTR          mx1.ipmac.lab.
```