5 Name Server and Configuration

5.1 DNS principles and Operations

DNS in Past

- Before DNS symbolic names to IP address mapping were stored in the file ‘hosts’ on each computer.
- This solution didn’t scale well because host file need to update on each computer every time a new host joins the network
- This file still exists(Contains static mappings)
    - Windows: c:\windows\system32\drivers\etc\hosts
    - Linux: /etc/hosts
- Problems with HOSTS.TXT
    - Consistency
    - Name collision

What is DNS?

- The Domain Name System is a distributed database with hierarchal structure and serve the basis for name resolution process in TCP/IP network.
- Domain Name System (DNS) converts the name of a Web site (www.bsccsitlab.com) to an IP address (202.42.34.12) and vice-versa.
- This IP is the IP address of a Web site's server, and is used in routing traffic over the Internet.

Benefits of DNS

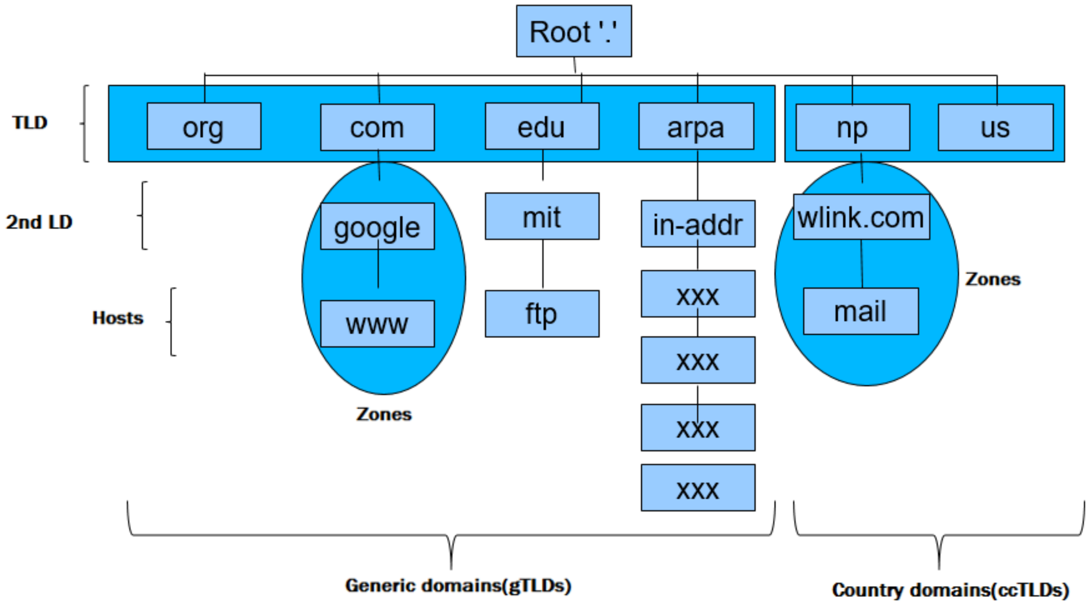
- DNS names are user-friendly, which means that they are easier to remember than IP addresses.
- DNS names remain more constant than IP addresses. An IP address for a server can change, but the server name remains the same.
- DNS allows users to connect to local servers using the same naming convention as the Internet

Components of DNS

- Domain namespace
- Zones
- Name servers
- Resolver

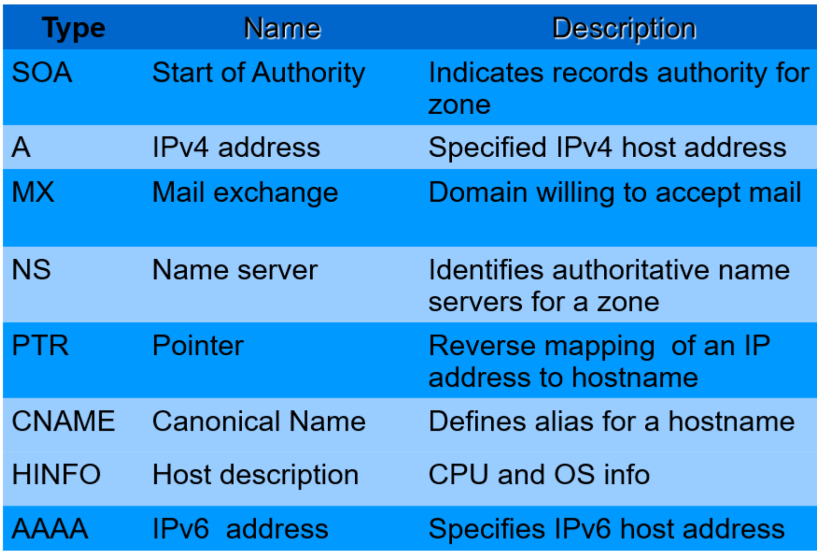
Domain name space

- DNS namespace is the hierarchical structure of the domain name tree.
- FQDN=www.google.com.



Zones

- All top-level domains, and many domains at the second and lower levels, are broken into zones.
- Zone is a part of domain name space that is smaller, more manageable units, separately administered.
- Each zone file consists of Domain name TTL, Class and type of resource records(RR)
- TTL defines how long the RR be cached
- Resource record is a standard DNS database structure containing information used to process DNS queries
- **Resource records**



Type	Name	Description
SOA	Start of Authority	Indicates records authority for zone
A	IPv4 address	Specified IPv4 host address
MX	Mail exchange	Domain willing to accept mail
NS	Name server	Identifies authoritative name servers for a zone
PTR	Pointer	Reverse mapping of an IP address to hostname
CNAME	Canonical Name	Defines alias for a hostname
HINFO	Host description	CPU and OS info
AAAA	IPv6 address	Specifies IPv6 host address

Name servers

- The servers that store information about the domain namespace are called name servers, ie. domain name to IP mapping and vice versa.
- The basic types of name servers are
  a. Root servers
     - Root servers are a name servers for the root zone of the domain name system of the internet
     - The root servers maintain a list of authoritative name servers that are responsible for each domain name and top-level domain.
     - There are 13 root name server specified with names in the form Letter.root-servers.net, where letter ranges from A to M
     - eg: a.root-servers.net, i.root-servers.net, m.root-servers.net

  b. Authoritative name server
     - An authoritative name servers is a name server that holds the actual DNS records
       - Primary name server (Master names server)
       - Secondary name server (Slave name server)

  c. Non-authoritative name server or Caching-only name server

Resolver

- Resolvers are the clients that access name servers. Programs running on a host that need information from the domain namespace use the resolver. The resolver handles:
  - Querying a name server.
  - Interpreting responses (which may be resource records or an error).
  - Returning the information to the programs that requested it.
- Some of tools used to query DNS are
  - dig
  - nslookup
  - host

DNS Operation and Query

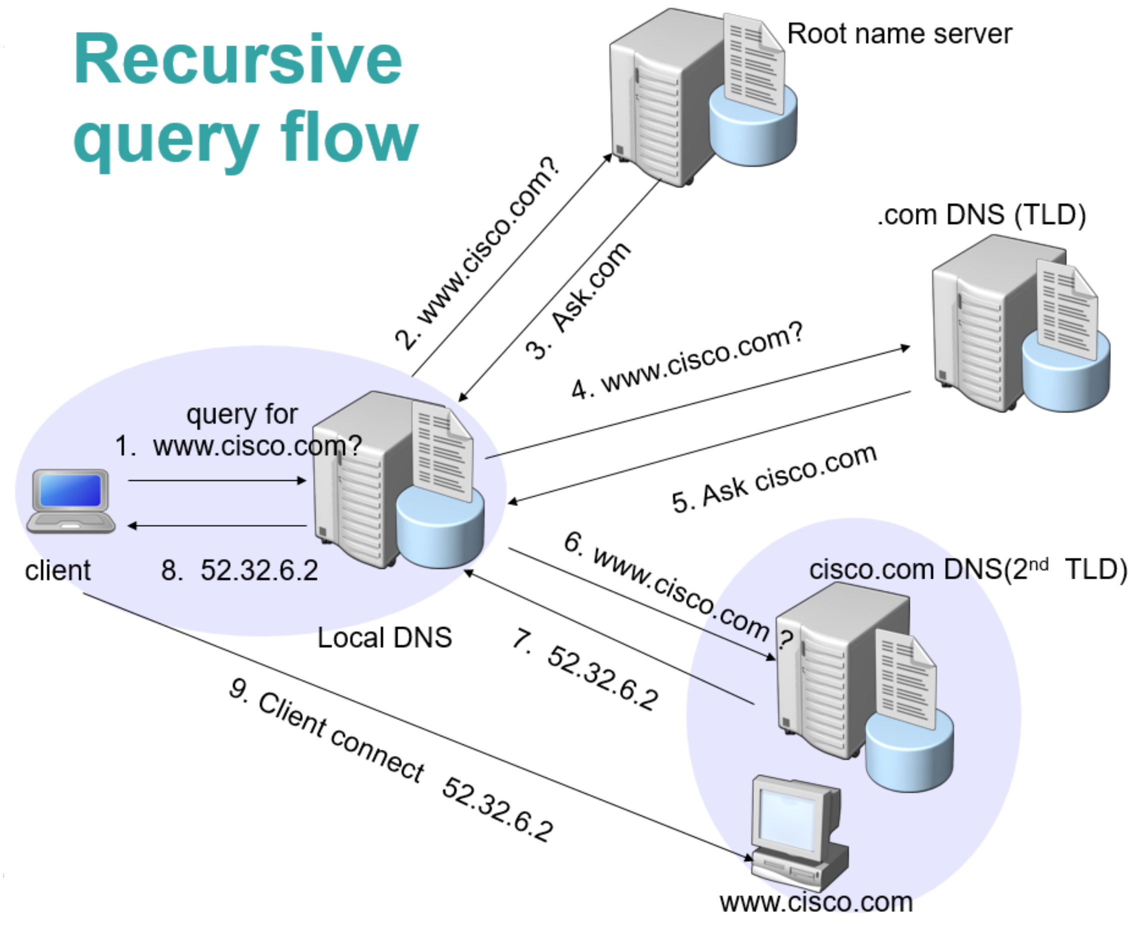
- DNS operation can be well explained with the help of DNS queries
- A query is a request for name resolution to a DNS server. There are two types of queries:
  - Recursive query
  - Iterative query

Recursive query

- A recursive query is a kind of query in which the local DNS server, who receives the query will do all the job of fetching the answer and reply the final answer back

- During the process, the local DNS server might also query other DNS server in the internet on the behalf for answer
- Let us understand the entire process of recursive query with the help of following block diagram.
- Suppose we want to browse www.cisco.com and our resolve.conf file has got the following entries

```
#less  /etc/resolve.conf
nameserver 192.168.1.10
nameserver 192.168.1.11
```



Recursive Query steps

Note:- Assume none of DNS server has contains in its cache

1. You enter ***www.cisco.com*** in the browser, so the operating system's resolver will send DNS query for the ***A record*** to the local DNS server 192.168.1.10
2. The local DNS server do not have the answer in its cache, so this server send s a query on the behalf of us to one of the root DNS server for the answer.
3. The root DNS server reply with the list of server that are responsible for handling the .com gTLD's.

4. The local DNS server will select one of the .com gTLD server from the list given by the root server to query for ***www.cisco.com***
5. The gTLD server replies back to the local DNS server with list of IP address of the authoritative DNS server that are responsible for the domain ***cisco.com***
6. The local DNS server will select one of the IP from the given list of authoritative DNS server and query for the ***A record*** of www.cisco.com
7. The authoritative name server will reply back to the local DNS server with ***A record*** as below

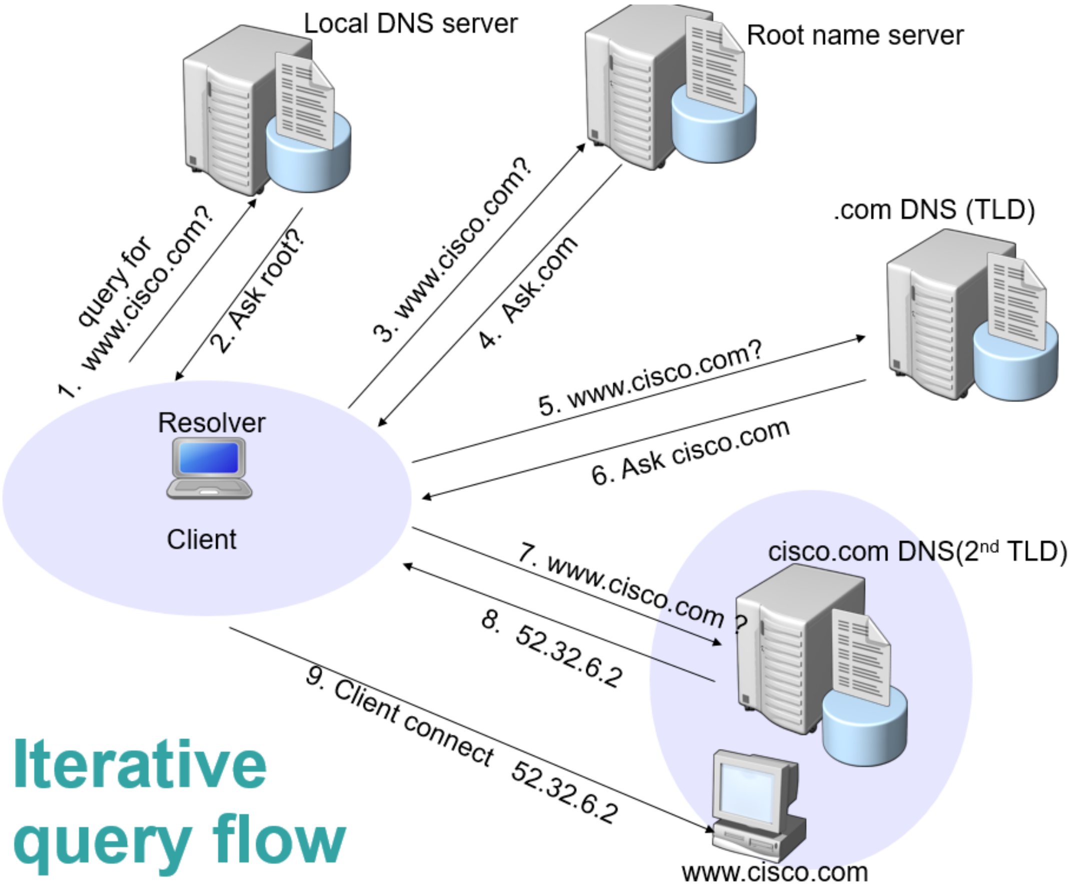
   www.cisco.com = 52.32.6.2

8. The local DNS server will reply back the answer to the client for ***A record*** of ***www.cisco.com***
9. Finally the browser will send request to the IP given for the web page of ***www.cisco.com***

Iterative query

- A iterative query is a kind of query in which the local DNS server will pass references to client and the client will do remaining queries itself.
- Let us understand the entire process of iterative query with the help of following block diagram.
- Suppose we want to browse www.cisco.com and our resolve.conf file has got the following entries

```
#less  /etc/resolve.conf
nameserver 192.168.1.10
nameserver 192.168.1.11
```



Iterative query steps

1. You enter ***www.cisco.com*** in the browser, so the operating system's resolver will send a DNS query for the ***A record*** to the local DNS server 192.168.1.10
2. The local DNS server does not have the answer in its cache, so instead of query the root server, it will reply us back with a referral to root servers.
3. Now our operating system resolver will query the root server for the answer.
4. The root DNS server reply with the list of server that are responsible for handling the .com gTLDs
5. The OS resolver will select one of the .com gTLD server form the list given by the root server to query the answer for ***www.cisco.com***
6. The gTLD server replies back to the resolver with the list of IP address of the authoritative DNS server that are responsible for the domain ***cisco.com***
7. The OS resolver will select one of the IP from the given list of authoritative DNS server and query for the ***A record*** of ***www.cisco.com***
8. The authoritative name server will reply back to the resolver with ***A record*** as below

    www.cisco.com = 52.32.6.2

9. Finally the browser will send request to the IP given for the web page of ***www.cisco.com***

5.2 Basic Name Server and Client Configuration

The basic name server configuration contains following parameter

- Main configuration file "/etc/named.conf" consists
  - Define listening port and interface
  - Allow subnets to query
  - Mention slave server for zone transfer
  - List authoritative zones (collection of records)
  - Other global configuration
    - Recursive query or not
    - dnssec enable or disable

- Client Configuration consists of nameserver defined under "/etc/resolv.conf
  nameserver 192.168.1.xxx (master IP)
  nameserver 192.168.1.xxx (slave IP)

5.3 Caching-only name server

- Caching-only name server stores (caches) DNS query results for a period of time determined in the configuration, TTL of each domain-name record.
- It improves the efficiency of DNS by reducing DNS traffic across the internet and by reducing load on authoritative name servers
- Caching-only DNS servers can improve security for your organization when used as forwarders that are under your administrative control.
- Internal DNS servers can be configured to use the caching-only DNS server as their forwarders and the caching-only DNS server performs recursion on behalf of your internal DNS servers.
- Sample configuration for caching-only name server in main configuration file

```
vi  /etc/named.conf
options {
        listen-on port 53 { 127.0.0.1 ; any ; };
        directory "/var/named";
        allow-query { localhost; any ; };
        recursion yes;
};
```

5.4 Primary and Slave Name Server

Authoritative name server

- An authoritative name servers is a name server that holds the actual DNS records (A, CNAME, PTR, MX..)
- There are basically two types of Authoritative name server

a. Master name server (Primary name server)
b. Slave name server (Secondary name server)

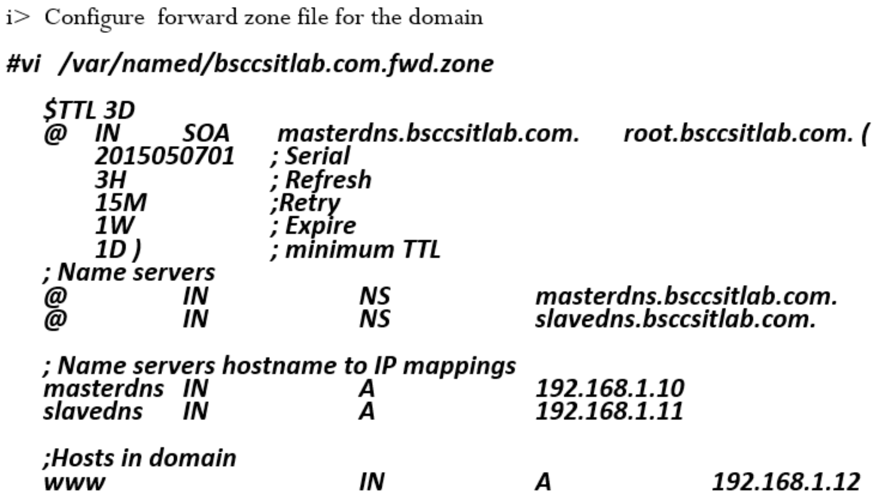
Master name server

- Master server stores the original master copies of all zone records.
- An administrator (host master) can only make changes and update to master server zone records.
- Each slave server gets updates via special automatic updating mechanism of DNS protocol
- Master may transfer full or partial zone updates based on the request from slave server.
- Sample configuration for Master name server consists of two parts
  a. Main configuration file
  b. Zone database configuration file

a. Main Configuration file for domain **bsccsitlab.com**

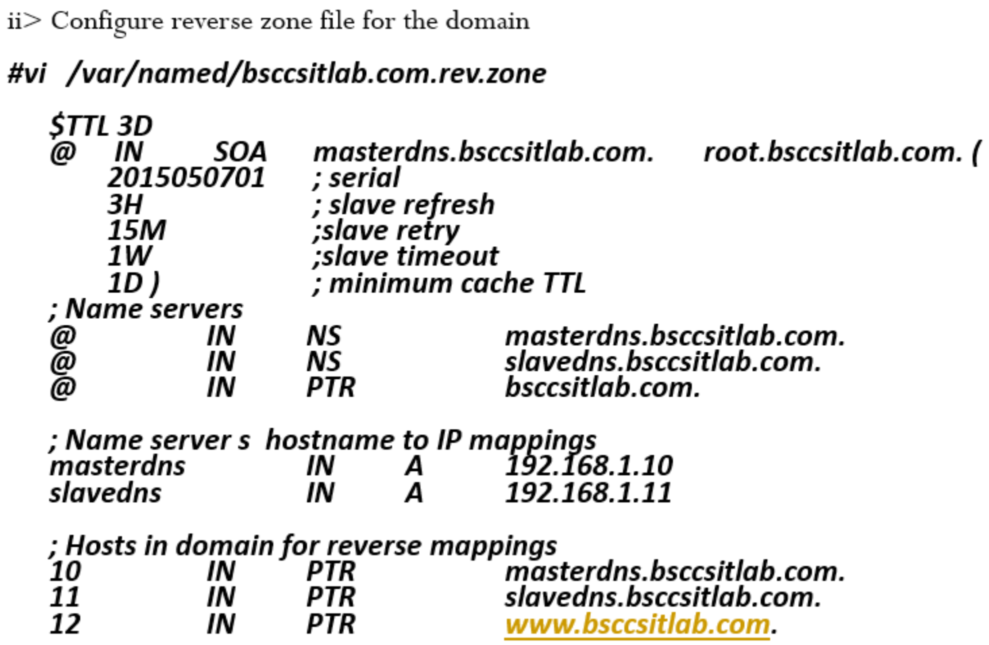
```
vi  /etc/named.conf
options {
          listen-on port 53 { 127.0.0.1 ; any ; };
          directory "/var/named";
          allow-query { localhost; any ; };
          allow-transfer{localhost; slave_IP;};
          recursion no;
          };
##Forward zone declaration#####
          zone “bsccsitlab.com” IN {
          type master;
          file “bsccsitlab.com.fwd.zone”;
          };
##Reverse zone declaration#####
          zone “1.168.192.in-addr.arpa” IN {
          type master;
          file “bsccsitlab.com.rev.zone”;
          };
```

b. Zone database configuration file for domain “bsccsitlab.com”



i> Configure forward zone file for the domain

```
#vi  /var/named/bsccsitlab.com.fwd.zone

  $TTL 3D
  @   IN      SOA     masterdns.bsccsitlab.com.      root.bsccsitlab.com. (
      2015050701      ; Serial
      3H              ; Refresh
      15M             ;Retry
      1W              ; Expire
      1D )            ; minimum TTL
  ; Name servers
  @           IN              NS              masterdns.bsccsitlab.com.
  @           IN              NS              slavedns.bsccsitlab.com.

  ; Name servers hostname to IP mappings
  masterdns   IN              A               192.168.1.10
  slavedns    IN              A               192.168.1.11

  ;Hosts in domain
  www                         IN              A                   192.168.1.12
```



ii> Configure reverse zone file for the domain

```
#vi   /var/named/bsccsitlab.com.rev.zone

     $TTL 3D
     @    IN      SOA     masterdns.bsccsitlab.com.     root.bsccsitlab.com. (
          2015050701      ; serial
          3H              ; slave refresh
          15M             ;slave retry
          1W              ;slave timeout
          1D )            ; minimum cache TTL
     ; Name servers
     @          IN      NS              masterdns.bsccsitlab.com.
     @          IN      NS              slavedns.bsccsitlab.com.
     @          IN      PTR             bsccsitlab.com.

     ; Name server s  hostname to IP mappings
     masterdns          IN      A       192.168.1.10
     slavedns           IN      A       192.168.1.11

     ; Hosts in domain for reverse mappings
     10         IN      PTR             masterdns.bsccsitlab.com.
     11         IN      PTR             slavedns.bsccsitlab.com.
     12         IN      PTR             www.bsccsitlab.com.
```

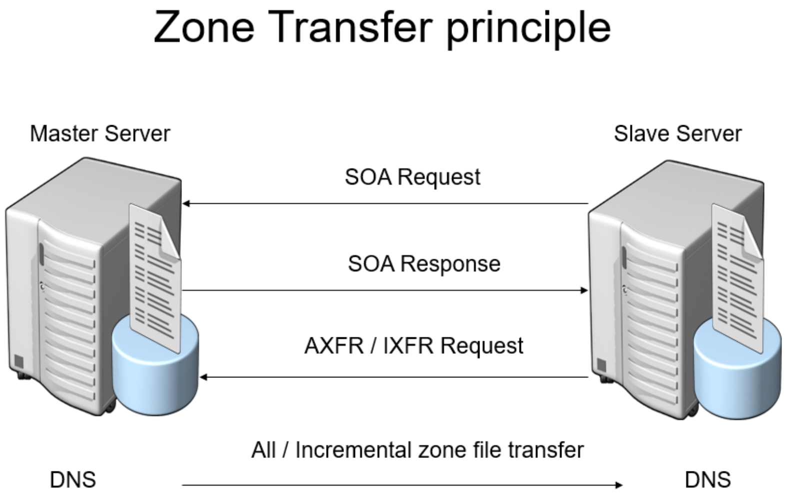
Slave name server

- Slave server is exact replica of master server
- It is used to share DNS server load and to improve DNS zone availability in case master server fails
- Each slave server gets updates via special automatic updating mechanism of DNS protocol
- Slave server initiate the zone file request, it could be full zone file transfer in the initial stage and partial zone file request on the subsequent updates.
- There could be more than one slave servers for a particular domain.
- Sample slave name server configuration

```
vi  /etc/named.conf
options {
            listen-on port 53 { 127.0.0.1 ; any ; };
            directory "/var/named";
            allow-query { localhost; any ; };
            recursion no;
            dnssec-enable no;
            dnssec-validation no;
};

##Forward zone declaration#####
 zone “bsccsitlab.com” IN {
            type slave;
            file “slaves/bsccsitlab.com.fwd.zone”;
            masters {192.168.1.10;};
};
##Reverse zone declaration#####
 zone “1.168.192.in-addr.arpa” IN {
            type slave;
            file “slaves/bsccsitlab.com.rev.zone”;
            masters {192.168.1.10;};
};
```

5.5 DNS Zone Transfer

- A zone transfer is the term used to refer to the process by which the contents of a DNS zone file are copied from a master DNS server to the slave DNS server.
- A zone transfer will occur during any of the following scenarios
  - When the refresh interval expires for the zone
  - When a slave server is notified of zone changes by its master server
  - When the DNS server service is started at a slave server for the zone.
- Zone transfer are always initiated by the slave DNS server, the primary DNS server simply answer the request for the zone transfer



- In this example the following sequence is performed for a requesting slave server to the master for a zone
  1. During new configurations, the slave server sends an initial "all zone" transfer (AXFR) request to the master server
  2. The master server response and fully transfer the zone to the slave server.
  3. When the refresh interval expires, and SOA request is used by slave server to renew the zone.
  4. The master response with its SOA record with the serial number for the zone in its current state at the master server.
  5. The slave server checks the serial number of the SOA record in the response and determines how to renew the zone
     a. if the value of the serial number in the SOA response is equal to its current local serial number, it concludes that the zone is the same at both servers and that a zone transfer is not needed.
     b. The slave server then renews the zone by resetting its refresh interval based on the value of this field in the SOA response from its master server.
     c. if the value of the serial number in the SOA response is higher than its current local serial number, it concludes that the zone has been updated and that a transfer is needed

6. If the slave concludes that the zone has changed, it sends ans IXFR request to the master.
7. The master server responds with either and incremental or full transfer of the zone.
    a. If the master server supports incremental transfer by maintaining a history of recent incremental zone changes for modified resource records, it can answer with an incremental zone transfer (IXFR) of the zone
    b. If the master server doesn't support incremental transfer or doesn't have a history of zone changes, it can answer with a full (AXFR) transfer of the zone

5.6 DNS Dynamic Updates

- ➢ Dynamic DNS is a system that addresses the problem of rapid updates. The term is used in two contexts which, while technically similar, have very different purposes and user populations.
- ➢ End users of Internet access receive an allocation of IP addresses, often only a single address, by their Internet service provider. The assigned addresses may either be fixed (or static), or may change from time to time, a situation called dynamic. Dynamic addresses are generally given only to residential customers and small businesses, as most enterprises specifically require static addresses.
- ➢ Dynamic IP addresses present a problem if the customer wants to provide a service to other users on the Internet, such as a web service. As the IP address may change frequently, corresponding domain names must be quickly re-mapped in the DNS, to maintain accessibility using a well-known URL.
- ➢ Many providers offer commercial or free Dynamic DNS service for this scenario. The automatic reconfiguration is generally implemented in the user's router or computer, which runs software to update the DDNS service. The communication between the user's equipment and the provider is not standardized, although a few standard web-based methods of updating have emerged over time.

5.7 DNS Delegation

- ➢ Delegation is very similar to how a manager will delegate responsibility of tasks to his staff. The results are the same, however more than one person was involved in the process
- ➢ DNS provides the option of dividing up the namespace into one or more zones, which can then be stored, distributed, and replicated to other DNS servers. When deciding whether to divide your DNS namespace to make additional zones, consider the following reasons to use additional zones:
    - A need to delegate management of part of your DNS namespace to another location or department within your organization.

- A need to divide one large zone into smaller zones for distributing traffic loads among multiple servers, improve DNS name resolution performance, or create a more fault-tolerant DNS environment.
- A need to extend the namespace by adding numerous sub-domains at once, such as to accommodate the opening of a new branch or site.

➢ For example: The following defines the hierarchy you want to create:

  zone domain name = bsccsit.com
  domain host name = www.bsccsit.com
  sub-domain name = acc.bsccsit.com
  sub-domain host name = ftp.acc.bsccsit.com

➢ To ease the administration load you want to fully **delegate** the responsibility for the administration of the ***acc*** sub-domain to the the ***acc.bsccsit.com*** management group, who puts these records on separate Authoritative DNS server.

5.8 Troubleshooting

Below are few useful troubleshooting commands

#named-checkconf

Command use to check "/etc/named.conf" for errors, the output should be empty.

#named-checkzone bsccsitlab.com /var/named/bsccsitlab.com.fwd.zone

Command used to check zone database files, the output should be as

```
"zone bsccsitlab.com/IN: loaded serial 2015050706
OK"
```

Check log files for DNS errors

#less /var/log/messages | grep named

DNS LAB:-

1. Caching-only name server
2. Master and Slave name server

Caching-only name server configuration

1) Package Required

- ***“bind-9.8.2-0.37.rc1.el6_7.7.i686”*** --main package for DNS server
- ***“bind-libs-9.8.2-0.37.rc1.el6_7.7.i686”*** --library packages for DNS
- ***“bind-utils-9.8.2-0.37.rc1.el6_7.7.i686”*** --utility package for DNS

2) Verify if the required package for DNS server is installed on your system or not by using following command

- ***#rpm –qa | grep bind***

3) If package does not exits, install it by using ***rpm*** command if package is locally available or by using ***yum*** if you are online

- ***#yum –y install bind***

4) DNS Server Configuration file

- ***“/etc/named.conf”*** -- Main configuration file
- **“*/var/named/”*** – Directory for DNS Zone files
- ***“/etc/init.d/named”*** -- script file for DNS
- ***“/etc/resolv.conf”*** --DNS client configuration file where name servers are defined

5) Backup Main configuration file

- ***#cp /etc/named.conf /etc/named.conf.backup***

6) Edit main configuration file

####Caching-only name server Configuration on 6th May 2016########

```
vi  /etc/named.conf
options {
        listen-on port 53 { 127.0.0.1 ; any ; };
        directory "/var/named";
        allow-query { localhost; any ; };
        recursion yes;
};
```

Save and exit
esc
:wq

7) Check and start DNS service

- ***#service named status*** --if named service is stopped, start is using following
- ***#service named start***

8) Enable service to start on system start-up/reboot

- ***#chkconfig --level 35 named on***

9) Set name server ip on “/etc/resolv.conf” file

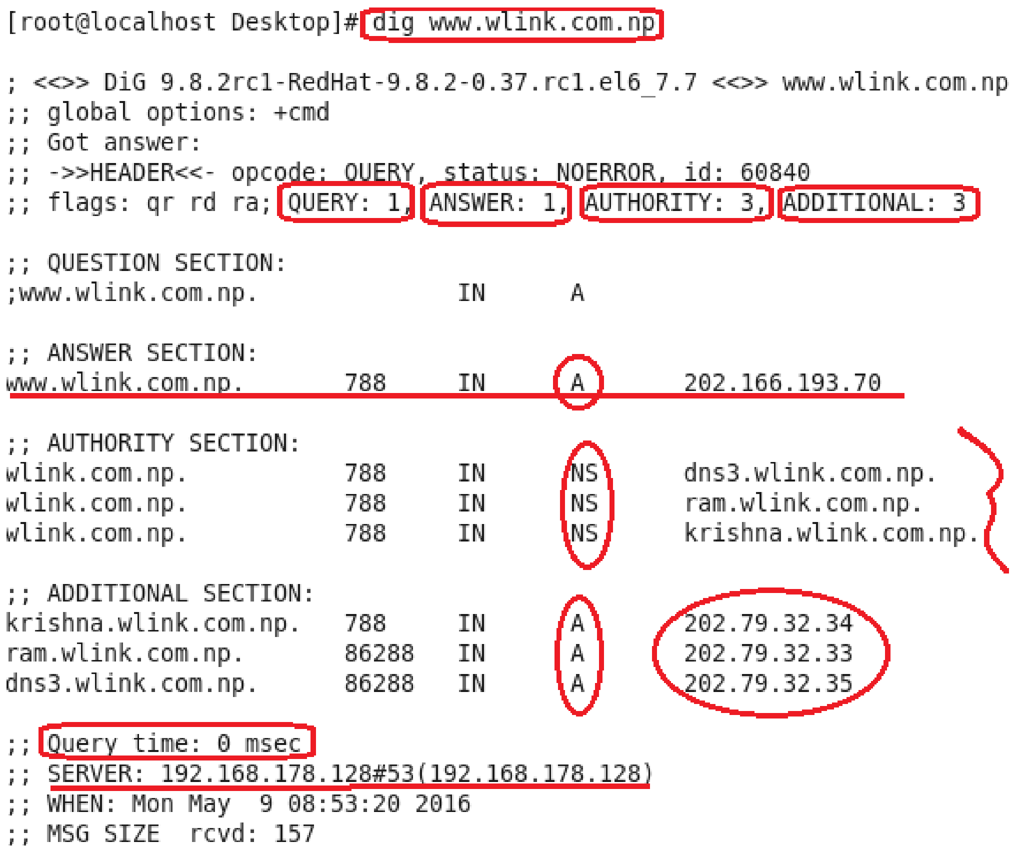
- ***#vi /etc/resolv.conf***
  nameserver 192.168.1.10

9) Test caching-only name server by using query tools such as

a. dig
b. nslookup
c. host

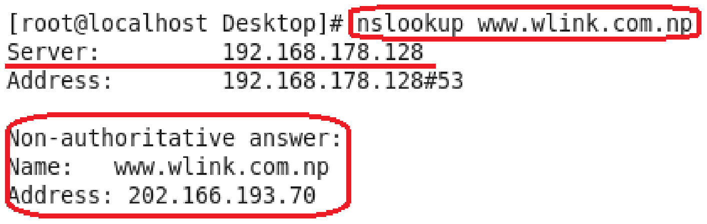
Example:-

a) dig www.wlink.com.np



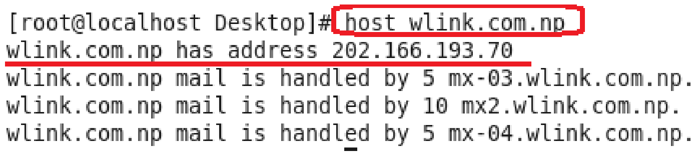
```
[root@localhost Desktop]# dig www.wlink.com.np

; <<>> DiG 9.8.2rc1-RedHat-9.8.2-0.37.rc1.el6_7.7 <<>> www.wlink.com.np
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 60840
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 3, ADDITIONAL: 3

;; QUESTION SECTION:
;www.wlink.com.np.              IN      A

;; ANSWER SECTION:
www.wlink.com.np.       788     IN      A       202.166.193.70

;; AUTHORITY SECTION:
wlink.com.np.           788     IN      NS      dns3.wlink.com.np.
wlink.com.np.           788     IN      NS      ram.wlink.com.np.
wlink.com.np.           788     IN      NS      krishna.wlink.com.np.

;; ADDITIONAL SECTION:
krishna.wlink.com.np.   788     IN      A       202.79.32.34
ram.wlink.com.np.       86288   IN      A       202.79.32.33
dns3.wlink.com.np.      86288   IN      A       202.79.32.35

;; Query time: 0 msec
;; SERVER: 192.168.178.128#53(192.168.178.128)
;; WHEN: Mon May  9 08:53:20 2016
;; MSG SIZE  rcvd: 157
```

b) nslookup www.wlink.com.np



```
[root@localhost Desktop]# nslookup www.wlink.com.np
Server:         192.168.178.128
Address:        192.168.178.128#53

Non-authoritative answer:
Name:   www.wlink.com.np
Address: 202.166.193.70
```

c) host www.wlink.com.np



```
[root@localhost Desktop]# host wlink.com.np
wlink.com.np has address 202.166.193.70
wlink.com.np mail is handled by 5 mx-03.wlink.com.np.
wlink.com.np mail is handled by 10 mx2.wlink.com.np.
wlink.com.np mail is handled by 5 mx-04.wlink.com.np.
```

Master and Slave name server configuration

A> Master Server Configuration for domain **bsscsitlab.com**

1) Package Required

- ***“bind-9.8.2-0.37.rc1.el6_7.7.i686”*** --main package for DNS server
- ***“bind-libs-9.8.2-0.37.rc1.el6_7.7.i686”*** --library packages for DNS
- ***“bind-utils-9.8.2-0.37.rc1.el6_7.7.i686”*** --utility package for DNS

2) Verify if the required package for DNS server is installed on your system or not by using following command

- ***#rpm –qa | grep bind***

3) If package does not exits, install it by using ***rpm*** command if package is locally available or by using ***yum*** if you are online

- ***#yum –y install bind***

4) DNS Server Configuration file

- ***“/etc/named.conf”*** -- Main configuration file
- ***“/var/named/”*** – Directory for DNS Zone files
- ***“/etc/init.d/named”*** -- script file for DNS
- ***“/etc/resolv.conf”*** --DNS client configuration file where name servers are defined

5) Backup Main configuration file

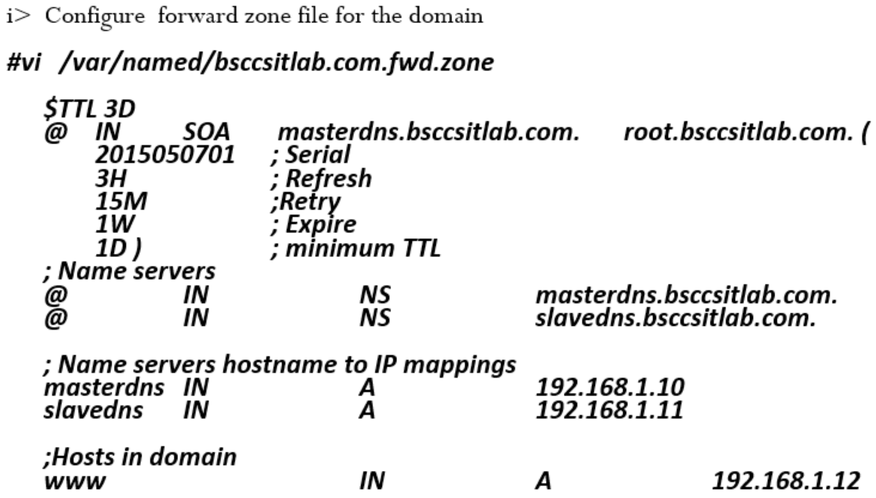
- ***#cp /etc/named.conf /etc/named.conf.backup***

6) Edit main configuration file

```
vi  /etc/named.conf
options {
            listen-on port 53 { 127.0.0.1 ; any ; };
            directory "/var/named";
            allow-query { localhost; any ; };
            allow-transfer{localhost; 192.168.1.11;};
            recursion no;
            };
##Forward zone declaration#####
            zone “bsccsitlab.com” IN {
            type master;
            file “bsccsitlab.com.fwd.zone”;
            };
##Reverse zone declaration#####
            zone “1.168.192.in-addr.arpa” IN {
            type master;
            file “bsccsitlab.com.rev.zone”;
            };
```

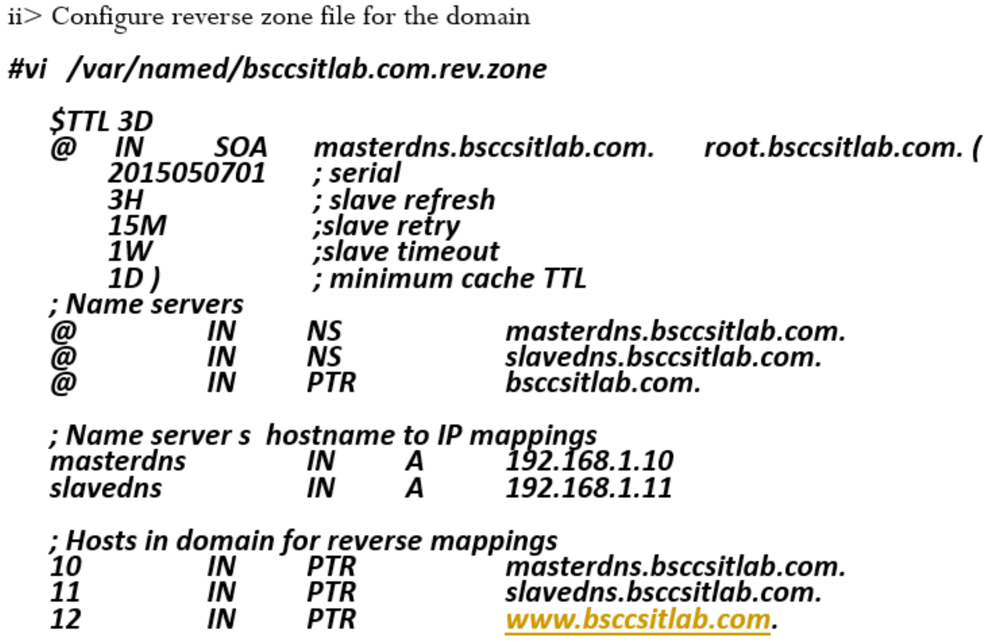
Save and exit
esc
:wq

7) Zone file configuration



i> Configure forward zone file for the domain

```
#vi  /var/named/bsccsitlab.com.fwd.zone

    $TTL 3D
    @   IN      SOA     masterdns.bsccsitlab.com.     root.bsccsitlab.com. (
        2015050701      ; Serial
        3H              ; Refresh
        15M             ;Retry
        1W              ; Expire
        1D )            ; minimum TTL
    ; Name servers
    @           IN                NS              masterdns.bsccsitlab.com.
    @           IN                NS              slavedns.bsccsitlab.com.

    ; Name servers hostname to IP mappings
    masterdns   IN                A               192.168.1.10
    slavedns    IN                A               192.168.1.11

    ;Hosts in domain
    www                           IN              A                 192.168.1.12
```



ii> Configure reverse zone file for the domain

```
#vi  /var/named/bsccsitlab.com.rev.zone

    $TTL 3D
    @    IN      SOA     masterdns.bsccsitlab.com.     root.bsccsitlab.com. (
         2015050701      ; serial
         3H              ; slave refresh
         15M             ;slave retry
         1W              ;slave timeout
         1D )            ; minimum cache TTL
    ; Name servers
    @              IN      NS              masterdns.bsccsitlab.com.
    @              IN      NS              slavedns.bsccsitlab.com.
    @              IN      PTR             bsccsitlab.com.

    ; Name server s  hostname to IP mappings
    masterdns              IN      A       192.168.1.10
    slavedns               IN      A       192.168.1.11

    ; Hosts in domain for reverse mappings
    10             IN      PTR             masterdns.bsccsitlab.com.
    11             IN      PTR             slavedns.bsccsitlab.com.
    12             IN      PTR             www.bsccsitlab.com.
```

Save and exit

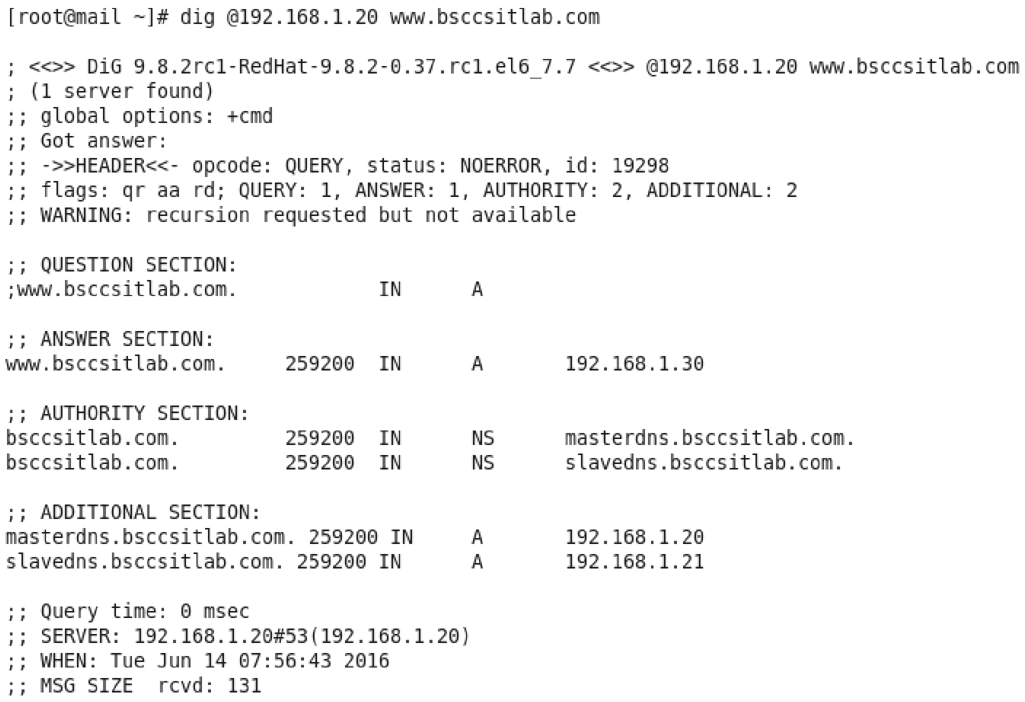
8) Check and start DNS service

- ***#service named status***   --if named service is stopped, start is using following
- ***#service named start***

9) Enable service to start on system start-up/reboot

- ***#chkconfig --level 35 named on***

10) Test master dns server by using dig command



```
[root@mail ~]# dig @192.168.1.20 www.bsccsitlab.com

; <<>> DiG 9.8.2rc1-RedHat-9.8.2-0.37.rc1.el6_7.7 <<>> @192.168.1.20 www.bsccsitlab.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 19298
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 2
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;www.bsccsitlab.com.            IN      A

;; ANSWER SECTION:
www.bsccsitlab.com.     259200  IN      A       192.168.1.30

;; AUTHORITY SECTION:
bsccsitlab.com.         259200  IN      NS      masterdns.bsccsitlab.com.
bsccsitlab.com.         259200  IN      NS      slavedns.bsccsitlab.com.

;; ADDITIONAL SECTION:
masterdns.bsccsitlab.com. 259200 IN     A       192.168.1.20
slavedns.bsccsitlab.com. 259200 IN      A       192.168.1.21

;; Query time: 0 msec
;; SERVER: 192.168.1.20#53(192.168.1.20)
;; WHEN: Tue Jun 14 07:56:43 2016
;; MSG SIZE  rcvd: 131
```

B> Slave Server Configuration

7)Edit main configuration file

```
vi  /etc/named.conf
            options {
                        listen-on port 53 { 127.0.0.1 ; any ; };
                        directory "/var/named";
                        allow-query { localhost; any ; };
                        recursion no;
                        dnssec-enable no;
                        dnssec-validation no;
            };

            ##Forward zone declaration#####
             zone “bsccsitlab.com” IN {
                        type slave;
                        file “slaves/bsccsitlab.com.fwd.zone”;
                        masters {192.168.1.10;};
            };
            ##Reverse zone declaration#####
             zone “1.168.192.in-addr.arpa” IN {
                        type slave;
                        file “slaves/bsccsitlab.com.rev.zone”;
                        masters {192.168.1.10;};
            };
```

Save and exit

8) Check and start DNS service

- ***#service named status*** --if named service is stopped, start is using following
- ***#service named start***

9) Enable service to start on system start-up/reboot

- ***#chkconfig --level 35 named on***

10) Check master and slave dns server by placing both dns server and client pcs on same network and edit "/etc/resolv.conf" file of client as below.

#vi /etc/resolv.conf

nameserver 192.168.1.10

nameserver 192.168.1.11

save and exit

esc

:wq

11)Now test using dig command on client PC and check whether master or slave server is responding.

Client#dig www.bsccsitlab.com

12) On Slave server verify if zone file is being transferred on its cache or not by using following command.

Slave# cat /var/named/slaves/bsccsitlab.com.fwd.zone

Slave# cat /var/named/slaves/bsccsitlab.com.rev.zone

You should see the zone database file being transferred from master DNS server.