DNS (Domain Name Server)

The DNS translates Internet domain and host names to IP addresses. DNS automatically converts the names we type in our Web browser address bar to the IP addresses of Web servers hosting those sites.

DNS implements a distributed database to store this name and address information for all public hosts on the Internet. DNS assumes IP addresses do not change (are statically assigned rather than dynamically assigned).

```
Hostname    to    192.168.1.2 (Called A Record)
192.168.1.2 to    hostname    (Called PTR Record)
Hostname    to    hostname    (Called CNAME Record)
```

In a DNS server such as BIND (Berkeley Internet Name Domain), all information is stored in basic data elements called resource records (RR). The resource record is usually a fully qualified domain name (FQDN) of a host, and is broken down into multiple sections organized into a tree-like hierarchy. This hierarchy consists of a main trunk, primary branches, secondary branches, and so on.

BIND consists of a set of DNS-related programs. It contains a nameserver called named, an administration utility called rndc, and a debugging tool called dig

How does DNS resolution work?

A client application requests an IP address from the name server usually by connecting to UDP port 53. The name server will attempt to resolve the FQDN based on its resolver library, which may contain authoritative information about the host requested or cached data about that name from an earlier query. If the name server does not already have the answer, it will turn to root name servers to determine the authoritative for the FQDN in question. Then, with that information, it will query the authoritative name servers for that name to determine the IP address.

What is an MX record?

An MX record numerically ranks the mail servers you would prefer to receive email for a domain. The MX record with the lowest number is preferred over the others, but you can set multiple email servers with the same value for simple load balancing.

When the named service is started, it reads the configuration from files as described in the following table

Path	**Description**
`/etc/named.conf`	The main configuration file.
`/etc/named/`	An auxiliary directory for configuration files that are included in the main configuration file

The process name of the service	=	named
Name of the rpm	=	bind

DNS Configuration Instructions:

- First check to see if you OS has already service rpm package installed on the system

```
rpm -qa | grep bind
bind-x.x.x.x.x86_64
bind-libs.x.x.x.x86_64
bind-utils.x.x.x86_64
rpcbind.x.x.x.x86_64     (x86_64 is for 64bit platform)
```

- If the above rpms exist then you can proceed with the configuration
- If the rpms do not exist then download the package/rpm using yum command

```
yum install bind -y
```

- Once the package is installed make changes to the following configuration file:

```
/etc/named.conf
```

Please Note: Additional parameter can be set by view more details at the bottom of this document

Example of a named.conf file on **primary nameserver**

```
zone "example.com" IN {
type master;
file "example.com.zone";
allow-transfer { 192.168.0.2; };
};
```

Example of a named.conf file on **secondary nameserver**

```
zone "example.com" {
type slave;
file "example.com.zone";
masters { 192.168.0.1; };
};
```

Where as

example.com	=	Your domain name
type master	=	Specify whether it is primary or secondary server
file example.com.zone	=	zone file located in /var/named
allow-transfer	=	If you want to transfer the zone entry to secondary server
masters	=	IP of master DNS server so the zone entries can be transferred

- Once you have the named.conf file configured then go to /var/named directory
- /var/named directory should have a few example zone files in it (e.g. named.localhost)
- Make a copy of example file and start creating zone files and add DNS records OR simply create a new file named example.com.zone based on the above setup and input the following parameters:

NOTE: Please make sure the new zone files you create has the correct ownership and group ownership as named

Example Zone File: (example.com.zone)

```
$TTL 1D
@ IN SOA hostname.example.com. root.example.com. (
      2001062501  ; serial
      21600       ; refresh after 6 hours
      3600        ; retry after 1 hour
      604800      ; expire after 1 week
      86400 )     ; minimum TTL of 1 day

;     Comments

      IN    NS    hostname.example.com.
      IN    NS    hostname2.example.com.
      IN    MX    20    hostname2.example.com.

$ORIGIN example.com
hostname     IN    A     192.168.1.30
london       IN    A     192.168.1.210
newyork      IN    A     192.168.1.211
lon          IN    CNAME london
```

Example Reverse File: (192.168.0.rev)

```
$TTL 1D
@     IN    SOA   hostname.example.com.   root.example.com.   (
      2001062501  ; serial
      21600       ; refresh after 6 hours
      3600        ; retry after 1 hour
      604800      ; expire after 1 week
      86400 )     ; minimum TTL of 1 day

;     Comments

        IN      NS      hostname.example.com.
        IN      MX      10      hostname.example.com.
        IN      MX      20      hostname2.example.com.

$ORIGIN 1.168.192.IN-ADDR.ARPA.
30              IN      PTR     hostname.example.com.
210             IN      PTR     london.example.com.
211             IN      PTR     newyork.example.com.
```

- Edit the /etc/hosts file to add your server so the NS parameter can be found

```
192.168.x.x yourhostname      yourhostname.example.com
```

- Edit the /etc/resolve.conf and add the following lines

```
domain      example.com
nameserver  DNS-IP
```

- Edit the /etc/nsswith.conf file and make sure the hosts entry is firs file then dns

```
hosts:      files dns
```

- Now start the httpd service

```
service named restart
```

- Also in order to have the service startup at the boot time run the following command

```
chkconfig named on
```

- Now run a query to search for a record

```
nslookup    dns1
```

```
OR
nslookup    dns1.example.com
```

Other DNS Related Information:

- **Start/Stop or Restart DNS**

  ```
  Systemctl restart named
  ```

- **Logs**

  `/var/log` = If set in named.conf file

- **Commands**

  ```
  ps -ef | grep named          =    List DNS daemon
  kill -1 pid                  =    Reset process whenever zone file
  ```

Using the rndc Utility

The rndc utility is a command line tool that allows you to administer the named service, both locally and from a remote machine. Its usage is as follows:

```
rndc [option...] command [command-option]
```

The rndc configuration is located in /etc/rndc.conf. If the file does not exist, the utility will use the key located in /etc/rndc.key, which was generated automatically during the installation process using the rndc-confgen -a command.

Checking the Service Status

To check the current status of the named service, use the following command:

```
# rndc status
```

To reload both the configuration file and zones, type the following at a shell prompt:

```
# rndc reload
```

This will reload the zones while keeping all previously cached responses, so that you can make changes to the zone files without losing all stored name resolutions.

To reload a single zone, specify its name after the reload command, for example:

```
# rndc reload localhost
```

Finally, to reload the configuration file and newly added zones only, type:

```
# rndc reconfig
```

Zone File Parameters Information:

$TTL

The $TTL directive allows you to set the default Time to Live (TTL) value for the zone, that is, how long is a zone record valid. Each resource record can contain its own TTL value, which overrides this directive.

Increasing this value allows remote nameservers to cache the zone information for a longer period of time, reducing the number of queries for the zone and lengthening the amount of time required to propagate resource record changes.

Example: Using the $TTL directive

```
$TTL 1D
```

Common Resource Records

The following resource records are commonly used in zone files:

A

The Address record specifies an IP address to be assigned to a name. It takes the following form:

```
hostname IN A IP-address
```

If the hostname value is omitted, the record will point to the last specified hostname.
"Using the A resource record", the requests for server1.example.com are pointed to 10.0.1.3 or 10.0.1.5.
Using the A resource record

```
server1 IN A 10.0.1.3 IN A 10.0.1.5
```

CNAME

The Canonical Name record maps one name to another. Because of this, this type of record is sometimes referred to as an alias record. It takes the following form:

```
alias-name IN CNAME real-name
```

CNAME records are most commonly used to point to services that use a common naming scheme, such as www for Web servers. However, there are multiple restrictions for their usage:

CNAME records should not point to other CNAME records. This is mainly to avoid possible infinite loops. CNAME records should not contain other resource record types (such as A, NS, MX, etc.). The only exception are DNSSEC related records (that is, RRSIG, NSEC, etc.) when the zone is signed. Other resource record that point to the fully qualified domain name (FQDN) of a host (that is, NS, MX, PTR) should not point to a CNAME record.

"Using the CNAME resource record", the A record binds a hostname to an IP address, while the CNAME record points the commonly used www hostname to it.

```
www IN CNAME server1
```

MX

The Mail Exchange record specifies where the mail sent to a particular namespace controlled by this zone should go. It takes the following form:

```
IN MX preference-value email-server-name
```

The email-server-name is a fully qualified domain name (FQDN). The preference-value allows numerical ranking of the email servers for a namespace, giving preference to some email systems over others. The MX resource record with the lowest preference-value is preferred over the others. However, multiple email servers can possess the same value to distribute email traffic evenly among them.

"Using the MX resource record", the first mail.example.com email server is preferred to the mail2.example.com email server when receiving email destined for the example.com domain.

```
example.com. IN MX 10 mail.example.com. IN MX 20 mail2.example.com
```

NS

The Nameserver record announces authoritative nameservers for a particular zone. It takes the following form:

```
IN NS nameserver-name
```

The nameserver-name should be a fully qualified domain name (FQDN). Note that when two nameservers are listed as authoritative for the domain, it is not important whether these nameservers are secondary nameservers, or if one of them is a primary server. They are both still considered authoritative.

Using the NS resource record

```
IN NS dns1.example.com.
IN NS dns2.example.com.
```

PTR

The Pointer record points to another part of the namespace. It takes the following form:

```
last-IP-digit IN PTR FQDN-of-system
```

The last-IP-digit directive is the last number in an IP address, and the FQDN-of-system is a fully qualified domain name (FQDN).

PTR records are primarily used for reverse name resolution, as they point IP addresses back to a particular name. Refer to Section 15.2.2.4.2, "A Reverse Name Resolution Zone File" for more examples of PTR records in use

SOA

The Start of Authority record announces important authoritative information about a namespace to the nameserver. Located after the directives, it is the first resource record in a zone file. It takes the following form:

```
@ IN SOA primary-name-server hostmaster-email (
      serial-number
      time-to-refresh
      time-to-retry
      time-to-expire
      minimum-TTL )
```

The directives are as follows:

- The @ symbol places the $ORIGIN directive (or the zone's name if the $ORIGIN directive is not set) as the namespace being defined by this SOA resource record.
- The primary-name-server directive is the hostname of the primary nameserver that is authoritative for this domain.
- The hostmaster-email directive is the email of the person to contact about the namespace.
- The serial-number directive is a numerical value incremented every time the zone file is altered to indicate it is time for the named service to reload the zone.
- The time-to-refresh directive is the numerical value secondary nameservers use to determine how long to wait before asking the primary nameserver if any changes have been made to the zone.
- The time-to-retry directive is a numerical value used by secondary nameservers to determine the length of time to wait before issuing a refresh request in the event that the primary nameserver is not answering. If the primary server has not replied to a refresh request before the amount of time specified in the time-to-expire directive elapses, the secondary servers stop responding as an authority for requests concerning that namespace.
- In BIND 4 and 8, the minimum-TTL directive is the amount of time other nameservers cache the zone's information. In BIND 9, it defines how long negative answers are cached for. Caching of negative answers can be set to a maximum of 3 hours (that is, 3H).

When configuring BIND, all times are specified in seconds. However, it is possible to use abbreviations when specifying units of time other than seconds, such as minutes (M), hours (H), days (D), and weeks (W). Following table, “Seconds compared to other time units” shows an amount of time in seconds and the equivalent time in another format.

Seconds compared to other time units

Seconds	**Other Time Units**
60	1M
1800	30M
3600	1H
10800	3H
21600	6H
43200	12H
86400	1D
259200	3D
604800	1W
31536000	365D

Example: Using the SOA resource record

```
@ IN SOA dns1.example.com. hostmaster.example.com. (
2001062501 ; serial
21600 ; refresh after 6 hours
3600 ; retry after 1 hour
604800 ; expire after 1 week
86400 ) ; minimum TTL of 1 day
```

A simple zone file:

```
$ORIGIN example.com.
$TTL 86400
@ IN SOA dns1.example.com. hostmaster.example.com. (
      2001062501 ; serial
      21600 ; refresh after 6 hours
      3600 ; retry after 1 hour
      604800 ; expire after 1 week
      86400 ) ; minimum TTL of 1 day

;     Comments
;     Comments

      IN    NS    dns1.example.com.
      IN    NS    dns2.example.com.

dns1  IN    A     10.0.1.1
dns2  IN    A     10.0.1.2
dns1  IN    CNAME server1
```

Some Important Questions Regarding DNS

Q:1 What does BIND Stands for ?
Ans: BIND stands for Berkeley Internet Name Domain.

Q:2 What is DNS Server and its fundamentals ?
Ans: The Domain Name System (DNS) is a hierarchical, distributed database. It stores information for mapping Internet host names to IP addresses and vice versa, mail routing information, and other data used by Internet applications. Clients look up information in the DNS by calling a resolver library, which sends queries to one or more name servers and interprets the responses. The BIND 9 software distribution contains a name server, named, and a resolver library, liblwres.

Q:3 What is the default port of BIND ?
Ans: The BIND server is accessed via the network on port 53. Both TCP and UPD ports are used. Queries are made via UDP & Responses are made via UDP unless the response is too large to fit in a single packet , If the response won't fit in a single UDP packet, then the response is returned via TCP.

Q:4 How will you define Domain Name ?

Ans: The data stored in the DNS is identified by domain names that are organized as a tree according to organizational or administrative boundaries. Each node of the tree, called a domain, is given a label. The domain name of the node is the concatenation of all the labels on the path from the node to the root node. This is represented in written form as a string of labels listed from right to left and separated by dots. A label need only be unique within its parent domain.
For example, a domain name for a host at the company Linuxtechi, Inc. could be mail.linuxtechi.com, where com is the top level domain to which mail.linuxtechi.com belongs, example is a subdomain of com, and 'mail' is the name of the host

Q:5 What are zone files in DNS server ?
Ans: The files which contain the data being served by the DNS system are called "Zone Files" They are made up of a series of "Resource Records". A Zone File will always contain an SOA record as well as additional records.

Q:6 What are the different types of DNS Server ?
Ans: Primary Master : The authoritative server where the master copy of the zone data is maintained is called the primary master server, or simply the primary. Typically it loads the zone contents from some local file edited by humans or perhaps generated mechanically from some other local file which is edited by humans. This file is called the zone file or master file.
Slave Server : The other authoritative servers, the slave servers (also known as secondary servers) load the zone contents from another server using a replication process known as a zone transfer. Typically the data are transferred directly from the primary master, but it is also possible to transfer it from another slave. In other words, a slave server may itself act as a master to a subordinate slave server.
Caching Name Server : Caching Name server is not authoritative for any zone, all queries are forwarded to other DNS servers if they are not stored in the DNS-cache zone. Answers for all queries are cached in DNS-cache zone for a time.
Forwarding : In this type of DNS server , all queries are forwarded to a specific list of name servers

Q:7 How the load balancing is achieved using DNS ?
Ans: A primitive form of load balancing can be achieved in the DNS by using multiple records (such as multiple A records) for one name. For example, if you have three WWW servers with network addresses of 10.0.0.1, 10.0.0.2 and 10.0.0.3, a set of records such as the following means that clients will connect to each machine one third of the time

multiple-a-records

When a resolver queries for these records, BIND will rotate them and respond to the query with the records in a different order. In the example above, clients will randomly receive records in the order 1,2, 3; 2, 3, 1; and 3, 1, 2. Most clients will use the first record returned and discard the rest.

Q:8 How to check syntax of named.conf is correct or not ?
Ans: named-checkconf is the command, which checks the syntax of named.conf file.

```
# named-checkconf /etc/named.conf
```

If bind is running in chroot environment use below command

```
# named-checkconf -t /var/named/chroot /etc/named.conf
```

Q:9 What are the different types of Resource Records in bind ?
Ans: Below are the list of resource records in bind :
SOA – start of authority, for a given zone
NS – name server
A – name-to-address mapping
PTR – address-to-name mapping
CNAME – canonical name (for aliases)
MX – mail exchanger (host to receive mail for this name)
TXT – textual info
RP – contact person for this zone
WKS – well known services
HINFO – host information
Comments start with ; continue to end of line

Q:10 Explain Bind chroot environment ?
Ans: Running bind in a chroot environment means named process will be limited to their directory only (/var/named/chroot). This can help improve system security by placing BIND in a ”sandbox”, which will limit the damage done if a server is compromised.

Q:11 What is domain delegation in Bind ?
Ans: Domain delegation means fully delegate the responsibility for a sub-domain to another name server.

Exmaple :

```
squid.linuxtechi.com      IN NS      ns2.linuxtechi.com
ns2.linuxtechi.com        IN A       192.168.1.51
```