

Ubuntu Server DNS

Primary DNS Server

Introduction

- What is a Primary DNS?
- What is an Authoritative DNS Lookup?
- What are zones?
- What are Zone file records
 - **SOA**
 - **A**
 - **MX**
 - **CNAME**
 - **Etc.**

Primary Master DNS

- The DNS server becomes authoritative for that zone.
- This Server manages the mappings for this zone and the answer will not be derived from anywhere else.
- You have spoken directly to the controlling server.

```
C:\Users\student>nslookup www.offcampusnetwork.co.uk  
Server:   ns.offcampusnetwork.co.uk  
Address:  192.168.100.1
```

```
Authoritative answer:  
Name:     www.offcampusnetwork.co.uk  
Address:  192.168.100.2
```

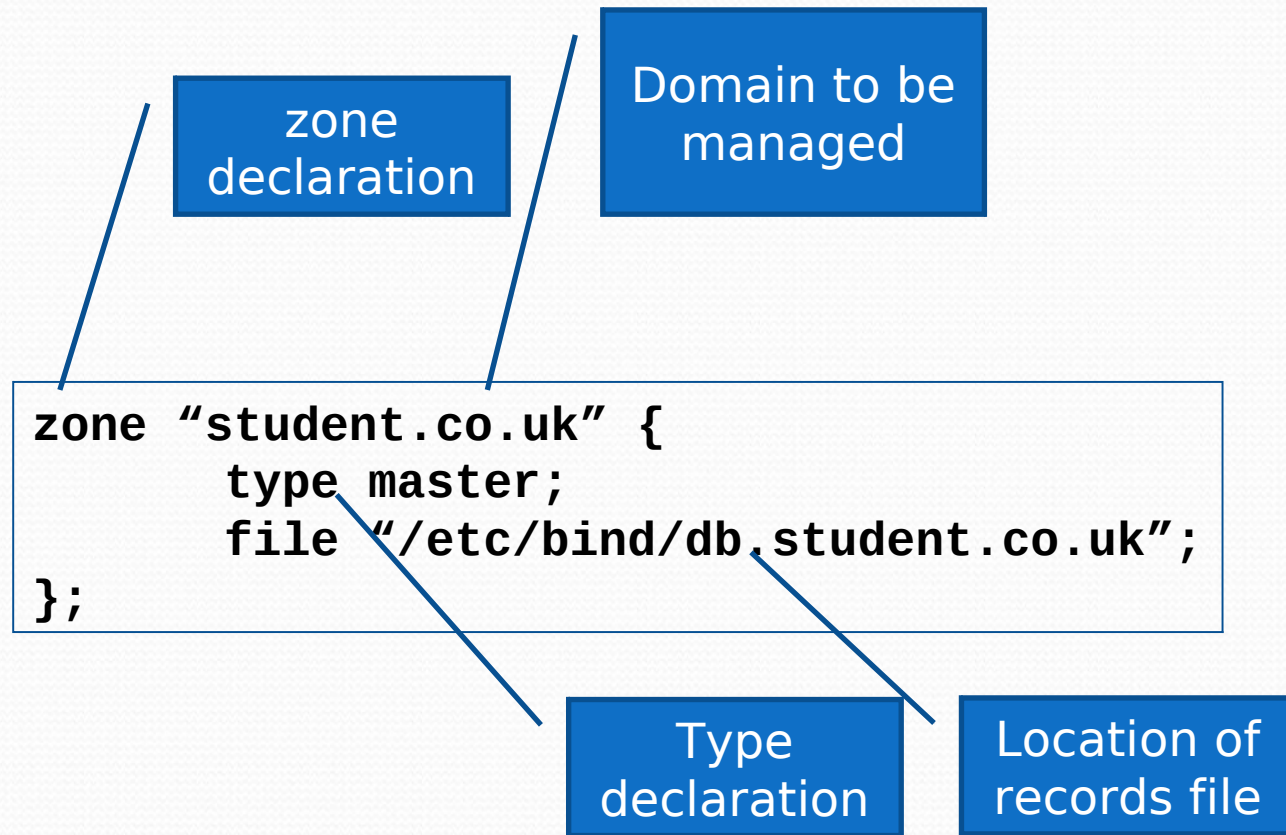

Configuring Primary Master DNS

- The configuration of a primary involves adding a zone.
- Zones are 'pointed to' from the **named.conf.local** file

```
/etc/bind/named.conf.local
```

- This **config** file allows you to specify the type of zone, e.g. **master**, and the location of its records, which are stored in a separate file.

Configuring Primary Master DNS



Configuring Primary Master DNS

Multiple zones can be declared and managed by one DNS

```
zone "student.co.uk" {  
    type master;  
    file "etc/bind/db.student.co.uk";  
};  
  
zone "academics.co.uk" {  
    type master;  
    file "etc/bind/db.academics.co.uk";  
};
```

Creating Zone Files

- Lets see how to create a zone file with the following information:
 - nameserver is **ns.student.co.uk**
 - Its IP address is **192.168.0.55**
 - Admin's email address is "**admin@student.co.uk**"
 - It has a www service running at the same IP address

Simple Zone File

```
;  
; BIND data file for student.co.uk  
;  
$TTL      604800  
@          IN      SOA      ns.student.co.uk. admin.student.co.uk. (  
                                3          ; Serial  
                                604800     ; Refresh  
                                86400      ; Retry  
                                2419200    ; Expire  
                                604800 )    ; Negative Cache TTL  
;  
; Name servers  
;  
@          IN      NS       ns.student.co.uk.
```


Simple Zone File ctd.

```
;  
; Addresses for the canonical names  
;  
localhost      IN      A      127.0.0.1  
ns             IN      A      192.168.0.55  
  
;  
; Aliases  
;  
www            IN      CNAME   ns.student.co.uk.
```

Creating Zone Files

- Serial Numbers
 - Many administrators like to use the last date edited as the serial of a zone
 - The format generally used is **yyyymmddss** (where *ss is the Serial Number*)
 - Eg. **2011100501**

Creating Zone Files

- When any changes are made to the Zone Files a service restart is required so the changes will be loaded.

```
sudo service bind9 restart
```


Zone File Format

- Zone files are text files (standardized by RFC 1035)
 - read or edited using any standard editor
- Contain three types of entries:
 - Comments
 - Directives
 - Resource Records

Zone File Format

- *Comments:* All comments start with a semicolon (;) and continue to the end of the line.
 - Comments can be added to any other record type and are assumed to terminate the line.
- *Directives:* All directives start with a dollar sign (\$) and are used to control processing of the zone files.
- *Resource Records:* Resource Records (RR) are used to define the characteristics, properties, or entities contained within the domain.
 - RRs are contained on a single line with the exception that entries enclosed in parentheses can spread across multiple lines.

Zone File Format

COMMENT

DIRECTIVE

RESOURCE
RECORD

```
; this is a full line comment
$TTL 12h ; directive - comment terminates the line
$ORIGIN example.com.
; Start of Authority (SOA) record defining the zone (domain)
; illustrates an RR record spread over more than one line
; using the enclosing parentheses
@ IN SOA ns1.example.com. hostmaster.example.com. (
    2011110800 ; se = serial number (COD DAY!)
    3h ; ref = refresh
    15m ; ret = update retry
    3w ; ex = expiry
    2h20m ; min = minimum
)
; single line RR
@ IN NS ns1.example.com.
```


Zone File Format

- The **\$TTL** directive
 - Defines the default Time to Live (TTL) value for the zone or domain, which is the time a RR may be cached (or saved) by another DNS server.
 - This directive is mandatory.
- The **\$ORIGIN** directive: The domain name for the zone being defined.
 - This directive is optional.

Zone File Format

- A *Start of Authority* (**SOA**) RR: The SOA RR, which must appear as the first RR in a zone file, describes the global characteristics of the zone or domain. There can be only one SOA RR in a zone file.
 - This RR is mandatory.
- The *Name Server* (**NS**) RR: Defines name servers that are authoritative for the zone or domain. There must be one or more (why?) NS RRs in a zone file.
 - NS RRs may reference servers in this domain or in a foreign or external domain.
 - These RRs are mandatory.

Zone File Format

- The *Mail Exchanger* (MX) RR: Defines the mail servers for the zone. There may be zero or more MX RRs in a zone file. If the domain does not provide e-mail services, there is no need for any MX RRs. An MX RR may reference a mail server in this domain or in a foreign or external domain.
 - This RR is optional.

Zone File Format

- The *Address (A)* RR:
 - Used to define the IPv4 address of all the hosts (or services) that exist in this zone and are required to be publicly visible.
 - IPv6 entries are defined using **AAAA** (called Quad A) RRs.
 - There may zero or more **A** or **AAAA** RRs in a zone file.
 - This RR is optional.

Zone File Format

- The **CNAME** RR: Defines an Alias RR, which allows one host (or service) to be defined as the alias name for another host.
 - There may be zero or more **CNAME** RRs in a zone file.
 - This RR is optional.

Zone File Format

```
; this is a full line comment
$TTL 12h ; directive - comment terminates the line
$ORIGIN example.com.
; Start of Authority (SOA) record defining the zone (domain)
; illustrates an RR record spread over more than one line
; using the enclosing parentheses
@ IN SOA ns1.example.com. hostmaster.example.com. (
    2011110800 ; se = serial number (COD DAY!)
    3h ; ref = refresh
    15m ; ret = update retry
    3w ; ex = expiry
    2h20m ; min = minimum
)
; single line RR's
    IN NS ns1.example.com.
    3w IN MX 10 mail.example.com.
ns1    IN A      192.168.254.2
mail   IN A      192.168.254.4
www    IN A      192.168.254.7
```


\$TTL

- Every Resource Record may take an optional Time to Live value specified in seconds.
- The **\$TTL** directive is standardized in RFC 2308 and defines the default TTL value applied to any RR that does not have an explicit TTL defined.
- TTL in the DNS context defines the time in seconds that a record may be cached by another name server or a *resolver*.
- Syntax
 - **\$TTL time-in-seconds**
 - e.g. **\$TTL 2d**
 - e.g. **\$TTL 172800**

\$TTL

- The time-in-seconds value
 - 0 - Indicates never cache the record,
 - 2147483647 - max, which is over 68 years
 - The current best practice recommendation (RFC 1912): RRs that rarely change should be given multi-week values.

\$TTL

- The **\$TTL** determines two DNS operational characteristics:
 - *Access load*: The lower the **\$TTL**, the more rapidly it is removed from resolver caches—forcing more frequent DNS queries to occur and thus raising the operational load on the zone's name server.
 - *Change propagation*: The **\$TTL** value represents the maximum time that any change will take to propagate from the zone name server to all users.

\$ORIGIN

- The **\$ORIGIN** directive was standardized in RFC 1035 it defines the domain name that will be appended to any incomplete name (sometimes called an *unqualified name*) defined in an *RR*.
- This process reduces the amount of work required to define machine names and addresses in *RR*.
- This process of appending to names that do not end with a dot is the major source of zone file configuration errors.

\$ORIGIN

- **The \$ORIGIN Substitution Rule**

- If a name appears in a RR and does not end with a dot, then the value of the last **\$ORIGIN** directive will be appended to the name. If the name ends with a dot, then it is a fully qualified domain name (FQDN) and nothing will be appended.

\$ORIGIN

- **Syntax**

- **\$ORIGIN domain-name**
- e.g.

\$ORIGIN example.com.

NOTE: FQDN ends
with a dot.

Zone Records

- **SOA**

- The first resource record (RR) in any Domain Name System Zone file should be a **Start Of Authority (SOA)** resource record.
- The **SOA** resource record indicates that this DNS name server is the best source of information for the data within this DNS domain.
 - **Authoritative**
- The **SOA** Resource Record defines the key characteristics and attributes for the zone or domain and is standardized in RFC 1035.

Zone Records

- SOA Syntax

- `name ttl class rr name-server e-mail sn
refresh retry expiry min`
- e.g.

Multi-line

```
@      IN SOA ns1.example.com. hostmaster.example.com. (  
        2011110800 ; sn = serial number  
        3h ; refresh time  
        15m ; retry = refresh retry  
        3w ; expiry  
        3h ; nx = nxdomain ttl  
)
```


Zone Records

- **SOA Syntax**

Syntax	Example	Description
name	@	The @ symbol substitutes the current value of \$ORIGIN (in the example file this is example.com.).
ttd		There is no ttd value defined for the RR (in this case), so the zone default of 2d (172800 seconds) from the \$TTL directive will be used.
class	IN	IN defines the class to be Internet (defaulted if omitted). Other values exist but are rarely used.
name-server	ns1.example.com.	Defines the Primary Master name server for the zone and has a special meaning only when used with Dynamic DNS configurations

Zone Records

● SOA Syntax

Syntax	Example	Description
e-mail	hostmaster.example.com.	Defines an administrative e-mail address for the zone. It is recommended in RFC 2142 that the e-mail address <i>hostmaster</i> be used uniquely for this purpose
sn	2011110800	Defines the serial number currently associated with the zone. The serial number <i>must</i> be updated every time any change is made to the domain. (usually using a date format?)
refresh	12h	When the refresh value is reached, the slave name server for this zone will try to read the SOA RR from the zone master

Zone Records

● SOA Syntax

Syntax	Example	Description
retry	15m	Defines the retry interval in seconds if the slave fails to make contact with the zone master during a refresh cycle
expiry	3w	Defines the time in seconds after which the zone records are assumed to be no longer authoritative. BIND interprets this to mean that the records can no longer be considered valid and consequentially stops responding to queries for the zone.
nx	3h	nx was redefined in RFC 2308 to be the period of time that negative responses can be cached by a resolver

Zone Records

- **SOA**

NOTE: This SOA is using Seconds to define all fields

```
@ IN SOA ns.example.com. root.example.com. (  
    2011110801      ; Serial  
    604800          ; Refresh  
    86400           ; Retry  
    2419200         ; Expire  
    604800 )        ; Negative Cache TTL
```

Note: if you update the zone file, you **MUST** update its serial number

Zone records

- **NS**

- The **NS** Resource Record is standardized in RFC 1035 and defines the authoritative name servers (there should be at least two) for the domain or zone
- A **NS** record tells name servers which machines are in charge of a given domain zone. This has one data-dependent field: The name of the DNS node which a given **NS** record points to.

Zone records

- **NS**
 - Syntax
 - `name ttl class rr name`
 - e.g.
 - `IN NS ns1.example.com.`

Zone records

- **NS Syntax**

Syntax	Example	Description
name		This field is blank (may be either a space or a tab character) and implicitly substitutes the current value of the name. You could also write this record as <code>example.com. IN NS ns1.example.com.</code>
ttl		There is no ttl value defined for the RR, so the zone default from the \$TTL directive will be used.
class	IN	Internet Class

Zone records

- **NS Syntax**

Syntax	Example	Description
name	ns1.example.com.	Defines a name server that is authoritative for the domain. In this example, an FQDN format has been used, but it could have been written as just ns1 (without the dot) and \$ORIGIN substitution would take place. This NS record points to a name server within the domain and therefore MUST have a corresponding A RR for IPv4 (or AAAA RR if IPv6) defined.

Zone Records

- **MX**
- The MX RR is standardized in RFC 1035 and defines the mail server(s) (*mail exchangers*) for the domain or zone.
 - Syntax
`name ttl class rr preference name`
 - e.g.
`3w IN MX 10 mail.example.com.`

Zone records

- MX Syntax

Syntax	Example	Description
name		This field is blank and implicitly substitutes the value of the right hand name field from the previous RR (in the example file, this is example.com.).
ttl	3w	This illustrates the use of an explicit ttl value in a RR that overrides the zone default (defined in the \$TTL directive). The value shown (three weeks) is significantly higher than the example zone default, which is two days. Because the domain MX RR is unlikely to change (its corresponding A RR may change more frequently)

Zone records

- MX Syntax

Syntax	Example	Description
class	IN	Internet Class
preference	10	The preference field indicates the relative preference or priority of the mail server it defines and can take any value between 0 and 65535. The lower the number, the more preferred the server.
name	mail.example.com.	Defines a mail server with the defined preference value for the domain. In this example, an FQDN format has been used, but you could write this as just mail

Zone records

- **A**

- The A RR is standardized in RFC 1035 and defines the IPv4 address of a particular host in the domain or zone.

- Syntax

`name ttl class rr ipv4`

- **e.g.**

- `ns1 IN A 192.168.254.2`
- `mail IN A 192.168.254.4`
- `www IN A 192.168.254.7`

Zone records

- **A Syntax**

Syntax	Example	Description
name	ns1	The name is unqualified, causing \$ORIGIN substitution. You could write this as ns1.example.com .
ttl		There is no ttl value defined for the RR, so the zone default from the \$TTL directive will be used. (0 would prevent caching!)
class	IN	Internet Class
ipv4	192.168.254.2	Defines that the host ns1 has the physical IPv4 address 192.168.254.2 . Records defined by NS or MX RRs that have names contained within this domain MUST have corresponding A RRs

Zone records

- **A**

- Returns a 32-bit **IPv4** address, most commonly used to map hostnames to an IP address of the host.

ns	IN	A	192.168.0.56
----	----	---	--------------

- **AAAA (Quad A)**

- Returns a 128-bit **IPv6** address, most commonly used to map hostnames to an IP address of the host.
 - We won't be covering this in this module other than DNS resolution

www	IN	AAAA	3ffe:1900:4545:2:02d0:09ff:fef7:6d2c
-----	----	------	--------------------------------------

Zone records

- **CNAME**

- The **CNAME** RR is standardized in RFC 1035 and defines an alias for an existing host defined by an **A** RR.

- Syntax

`name ttl class rr canonical-name`

- e.g.

`www IN CNAME server1.example.com.`

Zone Records

- CNAME Syntax

Syntax	Example	Description
name	www	The name is unqualified, causing \$ORIGIN substitution. You could write this as www.example.com .
ttl		There is no ttl value defined for the RR, so the zone default from the \$TTL directive will be used. (0 would prevent caching!)
class	IN	Internet Class
canonical -name	server1.example .com.	Defines that the name www.example.com is <i>aliased</i> to the host server1.example.com .

Zone Records

- **CNAME** Issues
 - **CNAME** RRs are often used when assigning service names to existing hosts.
 - If a host is actually called **server1** but runs an **ftp** and a **www** service, then CNAME RRs are frequently used to define these services.

ftp	IN	CNAME	server1
www	IN	CNAME	server1
server1	IN	A	192.168.254.21

Zone Records

- **CNAME** Issues

- It is permissible but considered very bad practice to chain **CNAME** records.

ns1	IN	A	192.168.254.2
server1	IN	A	192.168.254.21
www	IN	CNAME	mail
mail	IN	CNAME	server1


chaining
!



Zone Records

- **CNAME Issues**

- CNAME records should not be used with either NS or MX records



```
mail      IN MX      mail.example.com.  
mail IN CNAME www.example.com.  
www      IN A      192.168.254.7
```

The diagram illustrates a DNS configuration where a CNAME record for 'mail' points to 'www.example.com.', which in turn has an A record pointing to the IP address '192.168.254.7'. This setup is problematic because the 'mail' domain is also associated with an MX record pointing to 'mail.example.com.', creating a conflict as the CNAME record effectively masks the MX record.

Validating a Zone File

- The contents of a zone file can be complicated.
- There is a utility which will check a zone file's contents.
 - **named-checkzone**
- Syntax
 - **named-checkzone <domain> <zone file>**

Validating a Zone File

- e.g.

```
student@UbuntuServer:~$ named-checkzone  
student.co.uk /etc/bind/db.student.co.uk  
zone student.co.uk/IN: loaded serial 77  
OK  
student@UbuntuServer:~$
```


nslookup

- By default **nslookup.exe** does two lookups on a DNS server
 - A and **AAAA** records (Why?)
- You can set the option to only retrieve specific record types by using the **-type=?** Option
 - **-type=A**

```
C:\Users\student>nslookup -type=A www.hello.co.uk
Server:    UnKnown
Address:   192.168.0.55

Non-authoritative answer:
Name:      hello.co.uk
Address:   128.121.124.216
Aliases:   www.hello.co.uk
```


Conclusion

- What is a Primary DNS?
- What is a Primary DNS's main function?
- What is an Authoritative DNS Lookup?
- What is the advantage of having a Primary Server
- What is the advantage of having a Primary Caching Server with its own zones?