Module 6: Domain Name System (DNS)

Domain Name System (DNS)

6.1 DNS Fundamentals

- DNS is the system that translates domain names (like www.example.com) into IP addresses (like 192.0.2.1), allowing users to access websites and online services without needing to remember numerical addresses.

Key Concepts

- **Domain Names**: Human-readable names for websites and services.
- **IP Addresses**: Numerical labels assigned to devices on a network.

6.1 DNS Fundamentals

- **DNS Resolution**: The process of converting a domain name into its associated IP address.
- **Recursive and Authoritative DNS Servers**: Recursive servers find the correct authoritative DNS server for a domain, while authoritative servers store DNS records(Zone File)

6.2 DNS Servers & DNS Zones

- **DNS Servers**: The machines that respond to DNS queries. There are various types of DNS servers:

  1. Recursive DNS Servers: Resolve queries by contacting authoritative servers on behalf of clients.

  2. Authoritative DNS Servers: Provide the IP address corresponding to a requested domain name

  3. DNS TLD nameserver

  4. DNS root nameserver

- *We won’t go into details as this is not a networking course.*

6.2 DNS Servers & DNS Zones

- **DNS Zones**: A DNS zone is a portion of the DNS namespace managed by a specific entity. This includes the DNS records for all domains within that zone.
- **DNS Resolution process.**
- The DNS resolution process involves translating a domain name into an IP address. Here's how it works:

  1. **User query:** A user types www.example.com in the browser.

  2. **Recursive resolver**: The request is sent to a recursive DNS resolver, often provided by an ISP.

6.2 DNS Servers & DNS Zones

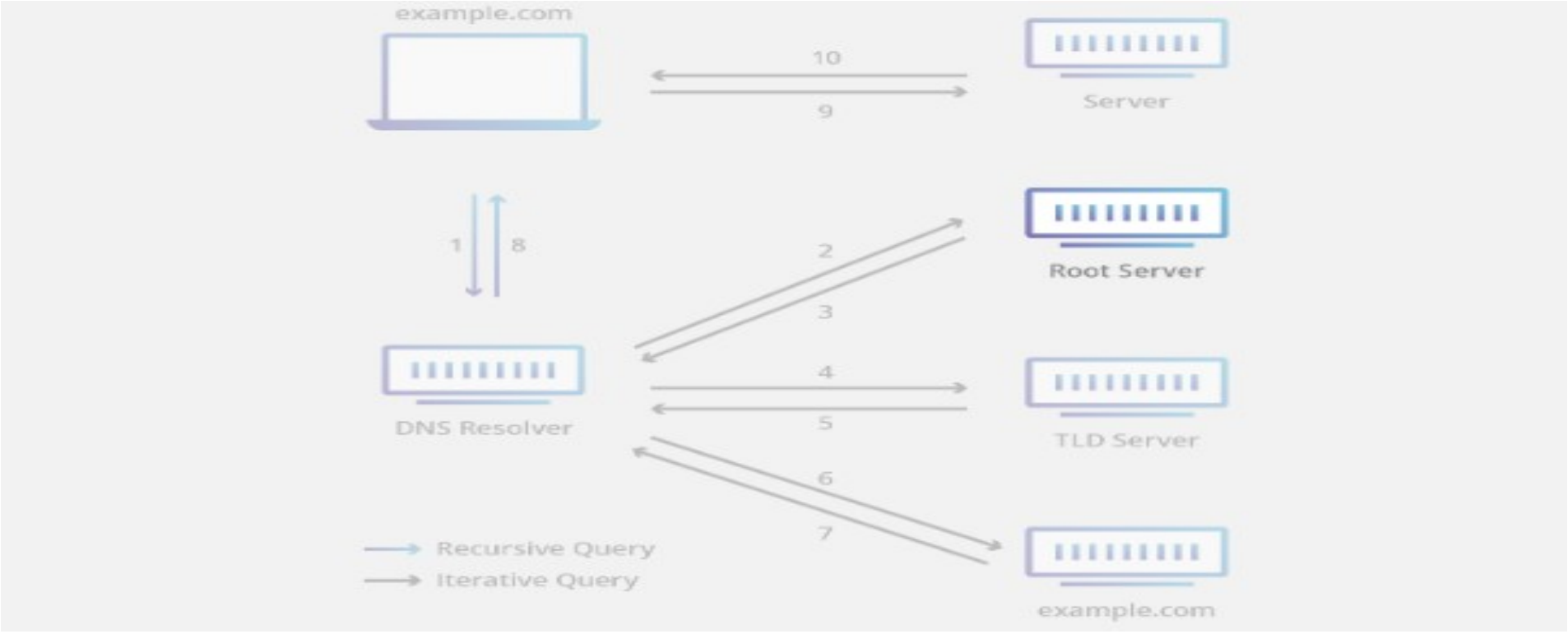
3. **Root server**: If the resolver doesn't have the IP cached, it queries a root DNS server.

4. TLD server: The root server directs the resolver to a TLD server, in our case, the .com server..

5. Authoritative DNS server: The TLD server directs the resolver to the authoritative DNS server for example.com

6. Response: The authoritative server responds with the IP address, which is returned to the user’s browser, allowing the site to load.

- 6.2 **DNS Resolution process : Illustration.**



6.3 Different DNS Record Types

- Each DNS record type serves a specific purpose, helping manage traffic for websites and services. Here are the most common types:

  1. A Record: Maps a domain name to an IPv4 address.

  2. **AAAA Record**: Maps a domain name to an IPv6 address.

  3. **CNAME Record**: Alias of one domain to another (e.g., www to example.com).

  4. MX Record: Mail exchange record used to route emails to the correct mail server.

6.3 Different DNS Record Types

5. TXT Record: Allows text-based data to be associated with a domain, often used for domain verification and email security (e.g., SPF, DKIM).

6. NS Record: Delegates a subdomain to another name server.

7. SOA Record: Start of Authority record, contains administrative information about the domain.

A (Address) Record:

- A (Address) Record:

Maps a domain name to its corresponding IPv4 address.

Example:

Name:	Type:	Value:
example.com	**A**	**192.0.2.1**

AAAA (IPv6 Address) Record:

- AAAA (IPv6 Address) Record:
-

Maps a domain name to its corresponding IPv6 address.

Example:

Name :	**Type:**	**Value:**
example.com	AAAA	2001:0db8:85a3:0000:0000:8a2e:0370:7334

CNAME (Canonical Name) Record:

Points a domain name to another domain name (alias).

Useful for setting up subdomains.

Example:

Name :	**Type:**	**Value:**
www.example.com	CNAME	example.com

MX (Mail Exchange) Record:

Specifies the mail server responsible for receiving email messages for the domain.

Includes a priority value to indicate the order of mail servers.

Example:

Name :	**Type:**	**Priority:**	**Value:**
example.com	MX	10	mail.example.com

TXT (Text) Record:

Allows domain administrators to insert text into the DNS.

Often used for verification purposes and to implement security policies such as SPF, DKIM, and DMARC.

Example :

Name :	**Type:**	**Value:**
example.com	TXT	"v=spf1 ip4:192.0.2.0/24 -all"

NS (Name Server) Record:

Specifies the authoritative name servers for the domain.

Example:

Name :	**Type:**	**Value:**
example.com	NS	ns1.example.com

PTR (Pointer) Record:

Maps an IP address to a domain name (reverse DNS lookup).

Commonly used in email server configurations.

Example:

Name :	**Type:**	**Value:**
1.2.0.192.in-addr.arpa	PTR	example.com

SOA (Start of Authority) Record:

Provides information about the domain's DNS zone, including the primary nameserver, admin email, serial number, and timers for refreshing the zone.

Example:

```
example.com SOA ns1.example.com. admin.example.com. (
                2023010101 ; serial
                7200       ; refresh (2 hours)
                3600       ; retry (1 hour)
                1209600    ; expire (2 weeks)
                86400      ; minimum (1 day)
```

6.4 Managing DNS

- Managing DNS effectively is crucial for the performance and reliability of your domain. This involves regularly reviewing DNS configurations, adding or updating DNS records, and ensuring that DNS zones are correctly set up.
- Key Tasks:

  1. **Adding and Updating DNS Records**: Whether it's setting up an A record for your website or configuring an MX record for email, DNS management tools allow you to create and update DNS records.

6.4 Managing DNS

2. **DNS managers** - Examples of platforms used to manage DNS are OLITT DNS manager, Cloudflare. Panels that support DNS management include cPanel, Plesk, Cyberpanel, DirectAdmin, CWP.

3. Some panels such as Cloudpanel do not currently support DNS management.

6.5 Configuring DNS Records

- To ensure that your website and services are correctly routed, it’s essential to properly configure DNS records. This includes setting up records like A, MX, and CNAME to point to the correct servers.
- Example Configurations:
- Setting an A record to point to a web server's IP address.
- Configuring MX records for your email provider.

6.6 TTL (Time to Live) and Propagation

- **TTL (Time to Live):** This is the time that a DNS record is cached by DNS resolvers before a new request is made to the authoritative server.

  Lower TTL: Faster propagation but higher query traffic to DNS servers.

  Higher TTL: Slower propagation but reduced query traffic.

- **Propagation**: After making DNS changes, it takes time for those changes to propagate across all DNS servers worldwide. This can take from a few minutes to 48 hours, depending on the TTL settings.

6.7 DNS Best Practices for Optimization

- Optimizing your DNS setup ensures faster response times and higher reliability for your websites and services.
- Best Practices:
- Use Multiple DNS Servers: Increase redundancy and improve resilience against failures by using multiple authoritative DNS servers.
- For this reasons, for example, Truehost uses multiple Authoritative DNS servers namely, ***ns1.cloudoon.com, ns2.cloudoon.net, ns3.cloudoon.org.***

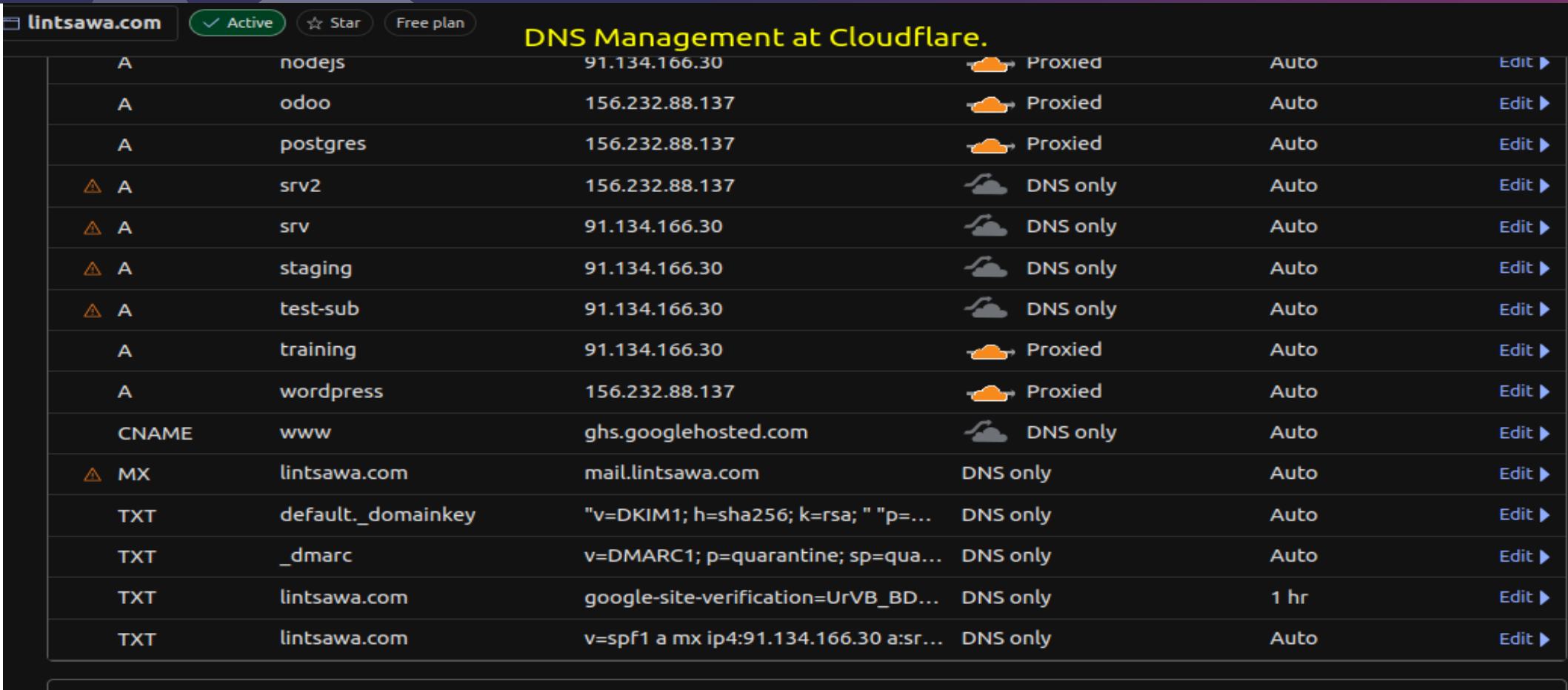
6.8 DNS Security

- Securing your DNS setup is critical to protecting your website and online services from attacks like DNS spoofing, cache poisoning, and DDoS attacks.
- **DNSSEC** (DNS Security Extensions): DNSSEC adds an additional layer of security by ensuring the authenticity and integrity of DNS data through digital signatures.

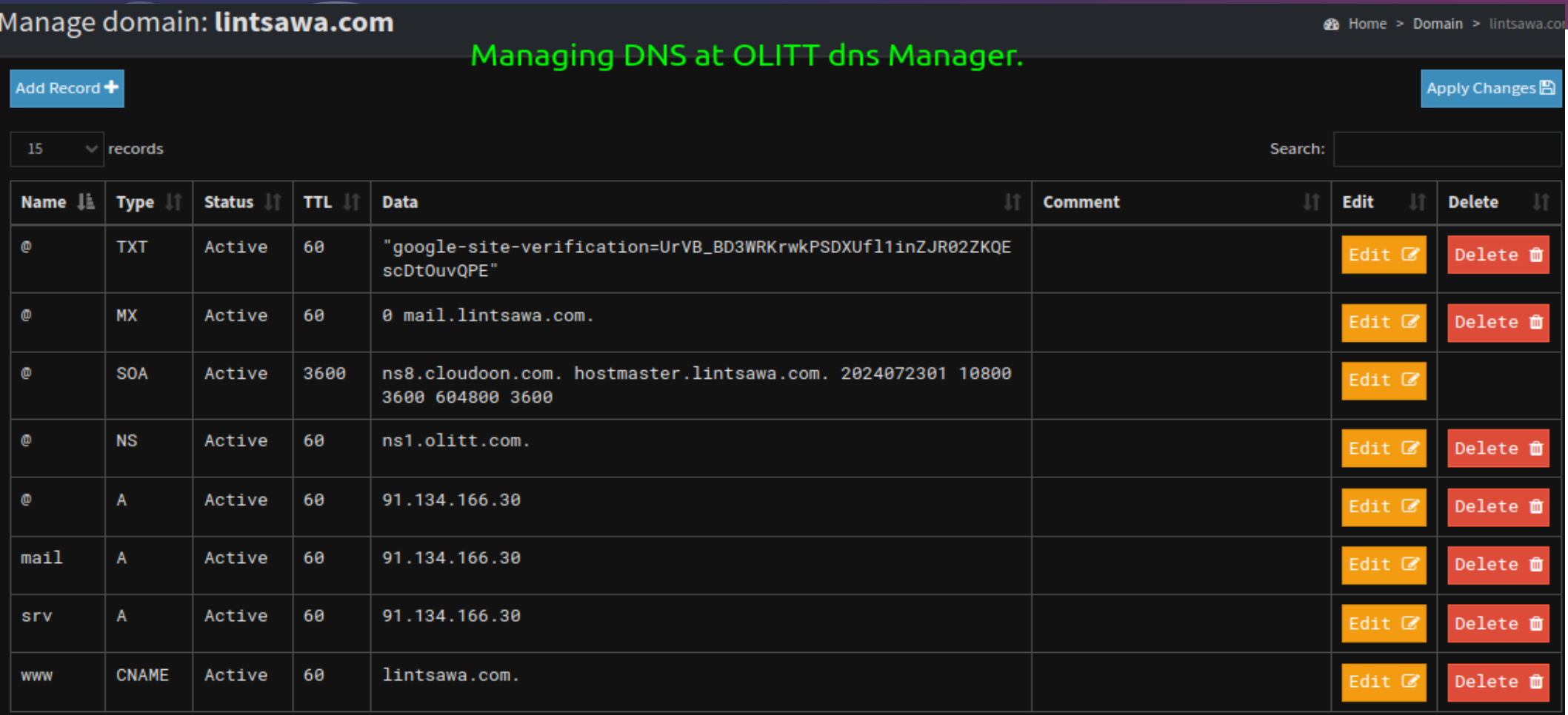
6.8 DNS Security

- Protecting Against DNS Attacks:
- DNS Spoofing: Implement DNSSEC to prevent attackers from redirecting traffic to malicious websites.
- DDoS Protection: Use services like Anycast DNS and cloud-based DNS providers to distribute traffic and prevent denial-of-service attacks such as Cloudflare.

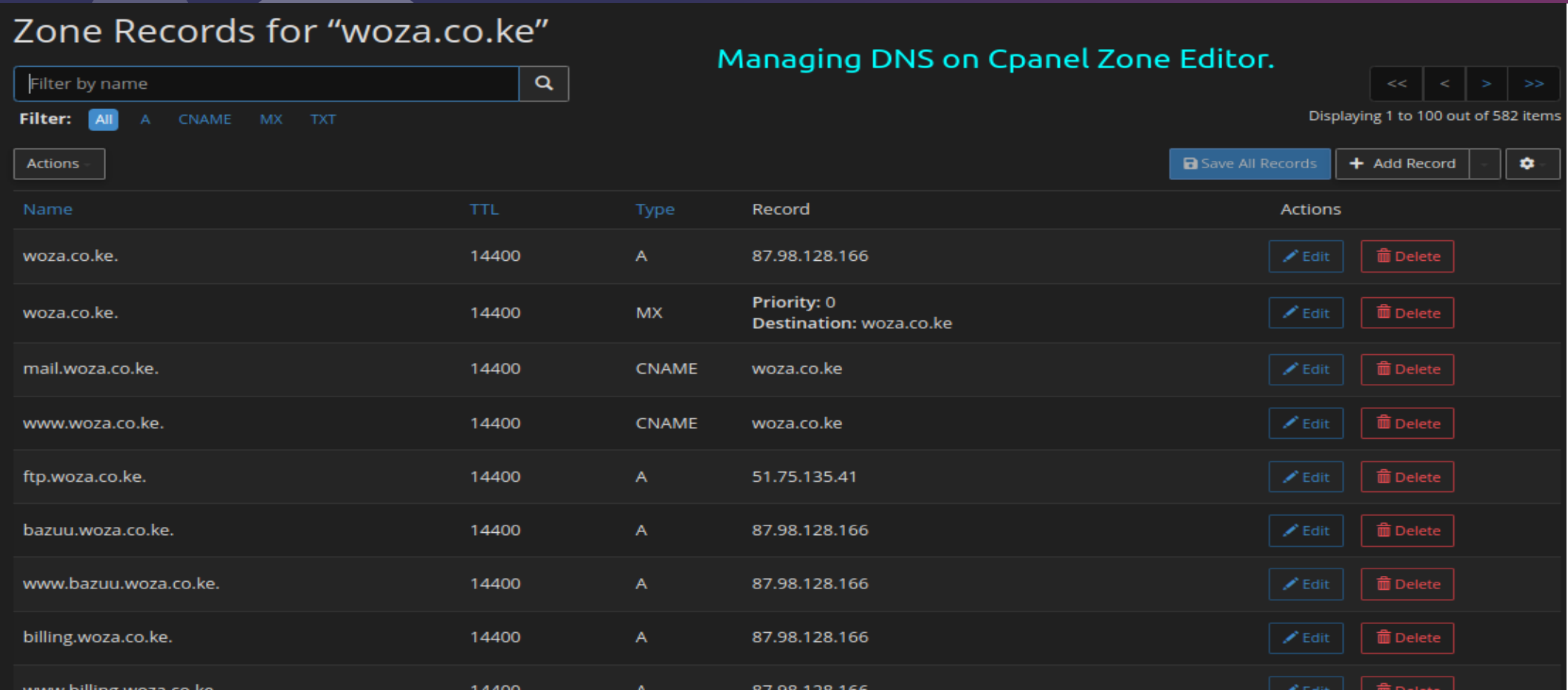
Illustrations



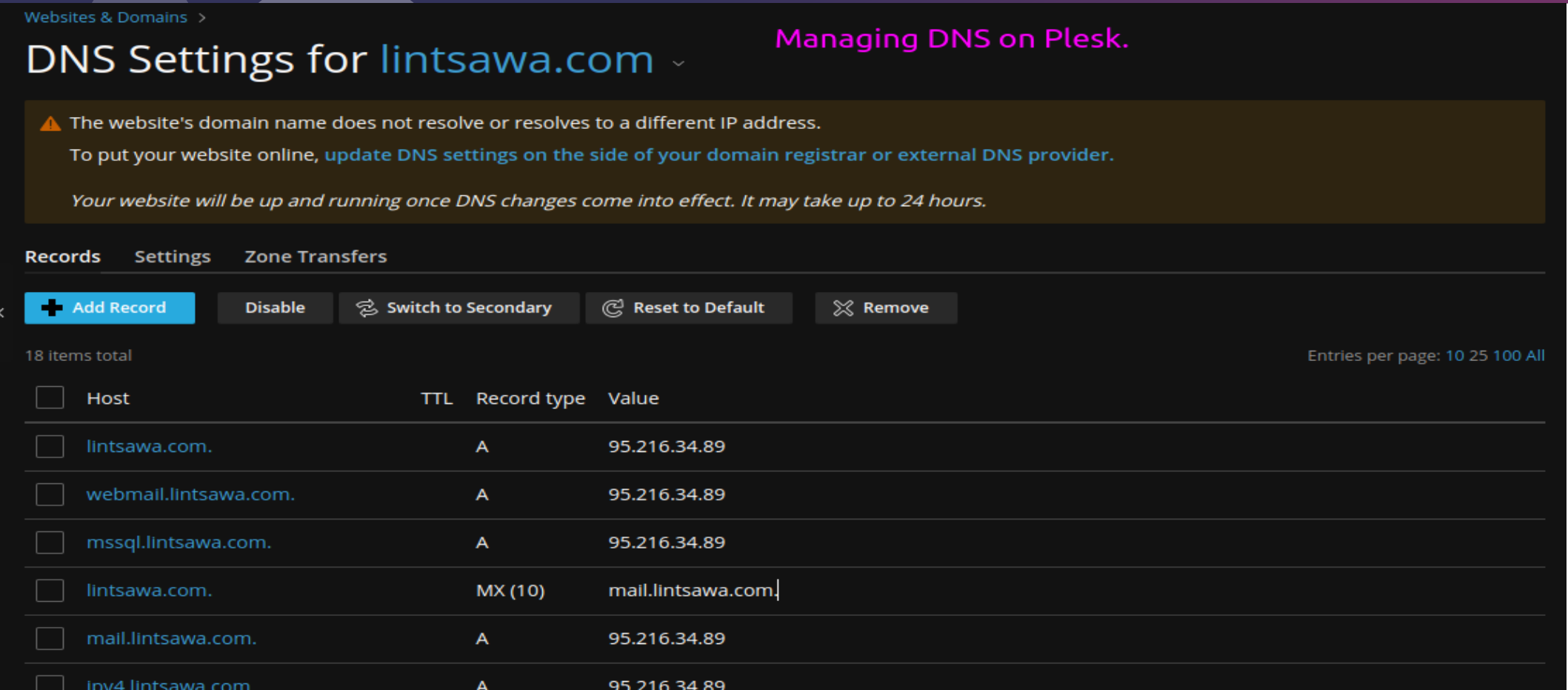
Illustrations



Illustrations



Illustrations



Illustrations