SELF-STUDY GUIDE – Load Balancer Solution With Nginx and SSL/TLS

This document explains the key technical concepts referenced in the main project setup.
It is intended for conceptual understanding and self-learning alongside the implementation.

---

1. The `/etc/hosts` File

What It Is:

The `/etc/hosts` file is a simple text file used by Linux systems for local hostname resolution.
It maps IP addresses to human-readable hostnames before DNS is queried.

Example:

```
172.31.20.187   Web1
172.31.28.1     Web2
```

Why It Matters:

- Allows your load balancer (Nginx) to resolve `Web1` and `Web2` locally without depending on DNS.
- Makes configurations simpler and readable.
- Essential when using private IP addresses in an AWS VPC.

2. DNS and Domain Records

DNS Basics:

DNS (Domain Name System) translates domain names (like `hairbydimani.store`) into IP addresses.

Types of Records:

- **A Record:** Maps a domain to an IPv4 address.
- **CNAME Record:** Maps one domain name to another.
- **MX Record:** Defines mail servers for the domain.

Why You Updated the A Record:

To point your domain name to the **Elastic IP** of your Nginx Load Balancer, so it becomes accessible via:

```
http://hairbydimani.store
https://hairbydimani.store
```

3. Nginx Load Balancing Concepts

What Load Balancing Does:

Distributes client requests across multiple servers (Web1 and Web2), improving:

- **Scalability** (handles more users)
- **Availability** (reduces downtime)
- **Performance** (balances workload)

Nginx Load Balancing Methods:

Method	Description	Use Case
Round Robin	Distributes requests sequentially	Default and simplest method
Least Connections	Sends requests to the server with the fewest active connections	Best for uneven workloads
IP Hash	Uses client IP for session persistence	Keeps users tied to the same backend

Why We Used Nginx:

- Lightweight and fast reverse proxy
- Handles both HTTP and HTTPS efficiently

- Simple SSL/TLS integration with Certbot

4. SSL/TLS and HTTPS

SSL/TLS Overview:

SSL (Secure Socket Layer) and **TLS (Transport Layer Security)** are cryptographic protocols that secure communication between clients and servers.

Why It’s Important:

- Encrypts data between the user and the website
- Ensures integrity and authentication
- Displays the “padlock” icon in browsers

How Let’s Encrypt and Certbot Work:

- **Let’s Encrypt:** Free certificate authority that issues SSL/TLS certificates.
- **Certbot:** A CLI tool that automates the process of obtaining and renewing these certificates.

Validation Process:

Certbot verifies that you control your domain (like `hairbydimani.store`) before issuing a certificate.

5. Elastic IP in AWS

What It Is:

An **Elastic IP (EIP)** is a static, public IPv4 address you can attach to your EC2 instance.

Why It’s Needed:

By default, EC2 public IPs change when you stop/start the instance.
An EIP ensures your Load Balancer’s IP **remains constant**, which is critical for DNS mappings.

6. Cron Jobs

What Is a Cron Job:

A **cron job** is a scheduled task in Unix/Linux systems that runs commands automatically at fixed intervals.

Example:

```
* */12 * * *   root /usr/bin/certbot renew > /dev/null 2>&1
```

This runs every 12 hours to renew your SSL certificate automatically.

Why It’s Important:

Ensures your HTTPS certificate never expires.
Manual renewals are error-prone — automation prevents downtime.

Summary

You should now understand:

- How Nginx distributes load between backend servers
- Why is Elastic IP essential for stable DNS
- How SSL/TLS protects web traffic
- The role of Certbot and cron jobs in maintaining continuous security

With this conceptual grounding, you can confidently manage a real-world load-balanced and secured web architecture.