



Paper 1

Introduction to Full Stack Web Development

Unit 1

How Websites Work

Module Leader

Sathishkumar Kannan, MS (UK)



2.1 Client-Server Architecture/Model

Clients:

- **Definition:** Clients are devices or software applications that request services or resources from a server.
- **Examples:** Web browsers (like Chrome, Firefox), mobile apps, or any device that accesses a network service.

Client Side (Web Browser):

- **Description:** The web browser is the client in this example.
- **Function:** It sends requests for web pages to a server and displays the received content.
- **Example Code:** HTML, CSS, and JavaScript in a simple webpage.

Servers:

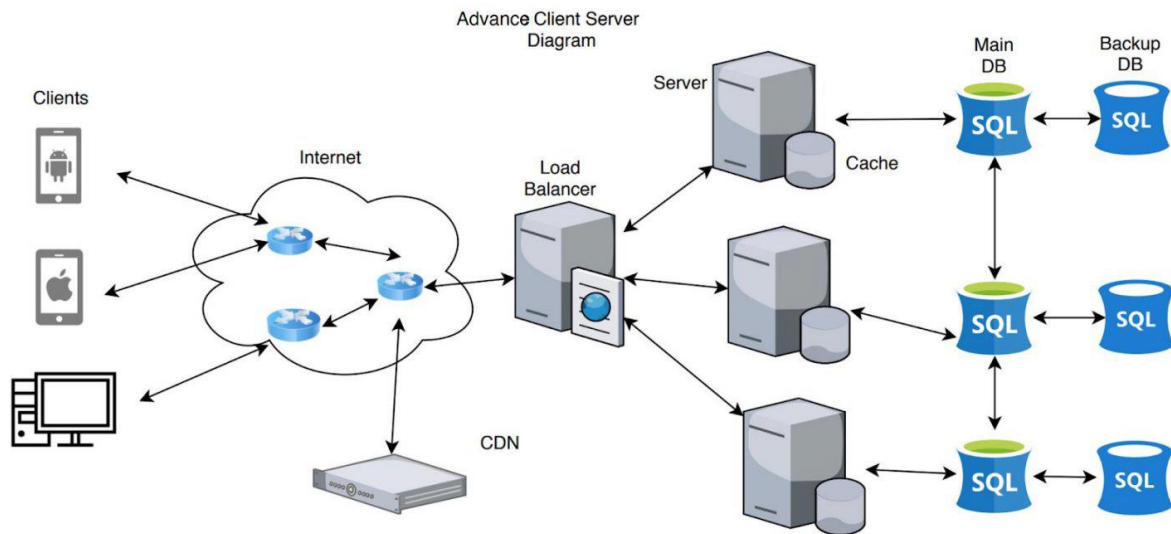
- **Definition:** Servers are powerful computers or software applications that provide services or resources to clients.
- **Examples:** Web servers (like Apache, Nginx), database servers (like MySQL, PostgreSQL), or application servers.

Server Side (Web Server):

- **Description:** The web server is the server in this example.
- **Function:** It receives requests from the client, processes them, and sends back the requested data.
- **Example Code:** A simple Node.js server that responds to the client's request.

Communication:

- Clients and servers communicate over a network using protocols (like HTTP, HTTPS, or TCP/IP) to exchange data.



Client: a piece of software or application that takes the input and sends request to the servers.

Server: a piece of software that receives and processes requests from clients.

Load balancer: responsible for distributing incoming network traffic across a group of backend servers to optimize resource usage

Network layer protocols such as TCP/IP

2.2 HTTP Protocol

HTTP (Hypertext Transfer Protocol) is the foundation of data communication on the World Wide Web. It defines how messages are formatted and transmitted.

2.3 Web browsers

Web browsers are applications that retrieve and present information resources on the World Wide Web. They interpret HTML, CSS, and JavaScript to display web pages.

- Open your favorite web browser (Chrome, Firefox, Safari, etc.).
- Enter a URL like "<https://www.example.com>" in the address bar.
- Press Enter, and the browser retrieves and displays the web page.

The browser acts as a client, making requests to the server to fetch and render web pages.



2.4 Request-Response Model

Clients send requests to servers, and servers respond to those requests with the required data or services.

Example

- The client (web browser) sends a request for a web page by entering a URL or clicking a link/button.
- The server (web server) processes the request, retrieves data if necessary, and sends it back to the client.
- The client displays the received data or renders the web page.
- This example illustrates the basic interaction between a client and a server in a client-server architecture. In a real-world scenario, the server could handle more complex tasks, such as retrieving data from a database, processing business logic, or interacting with other servers in the backend.

2.5 DNS and Domain Hosting

DNS (Domain Name System) translates human-readable domain names into IP addresses. Domain hosting involves storing website files on a server.

Example:

- **DNS Translation:**
 - Domain Name: www.example.com
 - IP Address: 192.168.1.1
- **Domain Hosting:**
 - Files for www.example.com are stored on a server.
 - The server's IP address is associated with the domain name.
- When you enter a domain in a browser, DNS resolves it to the corresponding server's IP.
- The browser contacts that IP to fetch and display the website.