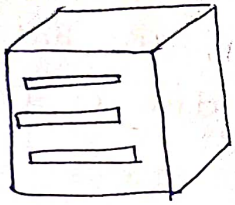
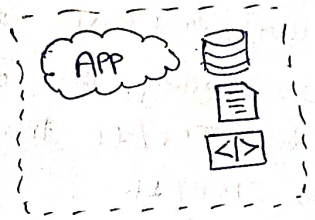
Episode -11 → Creating a Server

Server

- A server can be Referring to a hardware or software depending upon the context.
- A server is a computer or system that provides data, services, resources or program to other computers, called clients over a Network.
- When we buy a server on AWS, it is called an ec2 instance.
- We can use our own computer as a server too but there are limitations →
  1. We have limited storage and ram and expanding them takes time while AWS servers are scalable.
  2. Servers need to always up and run 24x7 which is not possible for an individual user.
  3. AWS computers have an dedicated IP while our IP may change.



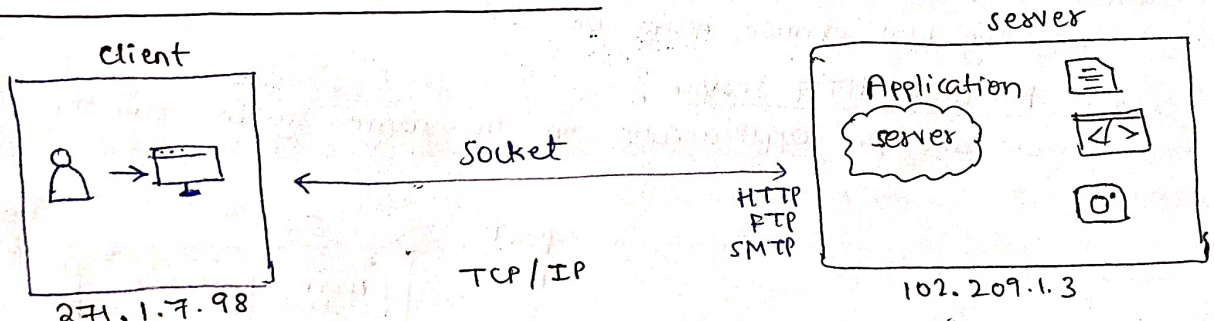
HARDWARE



SOFTWARE {server Application}

- We use NodeJS to create a server Application which handles user requests.

Client-Server Architecture



- Suppose a client wants to Access a file present on the server.
- The client has to open a socket connection and then request for file.
- The Application on server keeps listening to the requests and will accordingly gives the requested file to the client. and after this, the socket connection is closed.
- For Another request, the same process happens again, opening socket connection, making request, getting requested data and the closing of socket connection.

- When socket connection is made, the data is sent using TCP/IP protocol.

TCP/IP → protocols/rules which are followed for sending data on web.

HTTP ~~server~~, FTP, SMTP ⇒

- When client makes a socket connection to a server, there are different protocols/rules by which the server will sends its response in.

HTTP → Hyper Text Transfer Protocol { for sending simple text, html, json data }

FTP → File Transfer Protocol

SMTP → Simple Transfer Protocol

- Web server is like HTTP server, it defines rules how client talks to server
- The server keeps waiting for requests and modifies the response according to request and gives it back to client.

#Packets

- The data that comes client (web browser) from server comes in form of smaller chunks called Packets.

Note: The language that client and server speak is defined by HTTP, FTP, SMTP

TCP/IP is the protocol for sending the data over an IP, the data transmission is controlled by TCP/IP that is sending of data in form of packets is controlled by TCP/IP.

#Domain Names And DNS (Domain Name system)

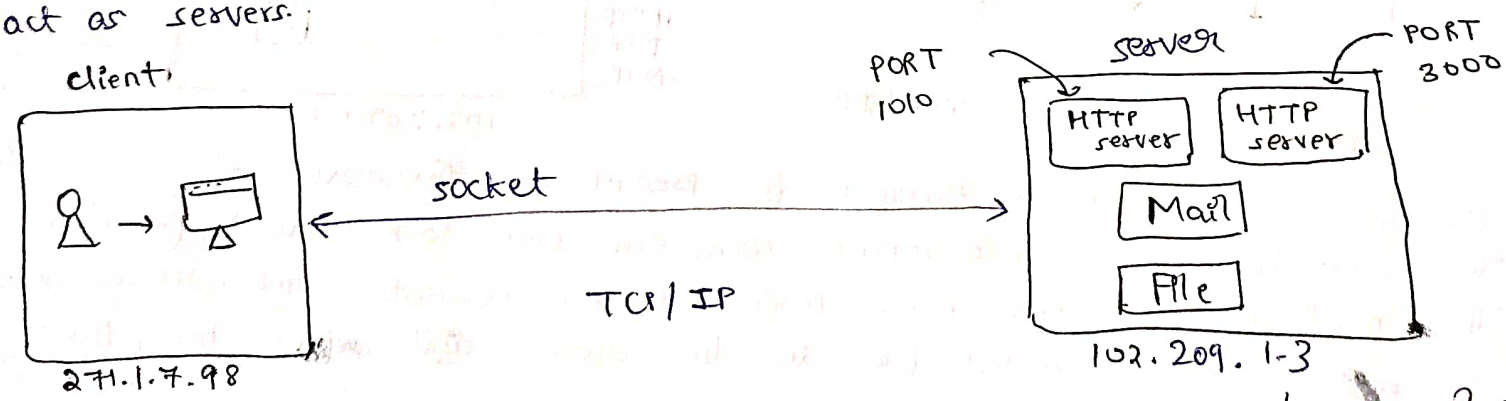
↳ it is a string of text that is mapped to an IP Address.

DNS → it turns domain names into IP Addresses, which allows browsers to get to websites and other internet resources.

→ manages mapping b/w domain name and IP.

Can we create Multiple HTTP Server?

Yes, we can create multiple applications on the same system and these will act as servers.

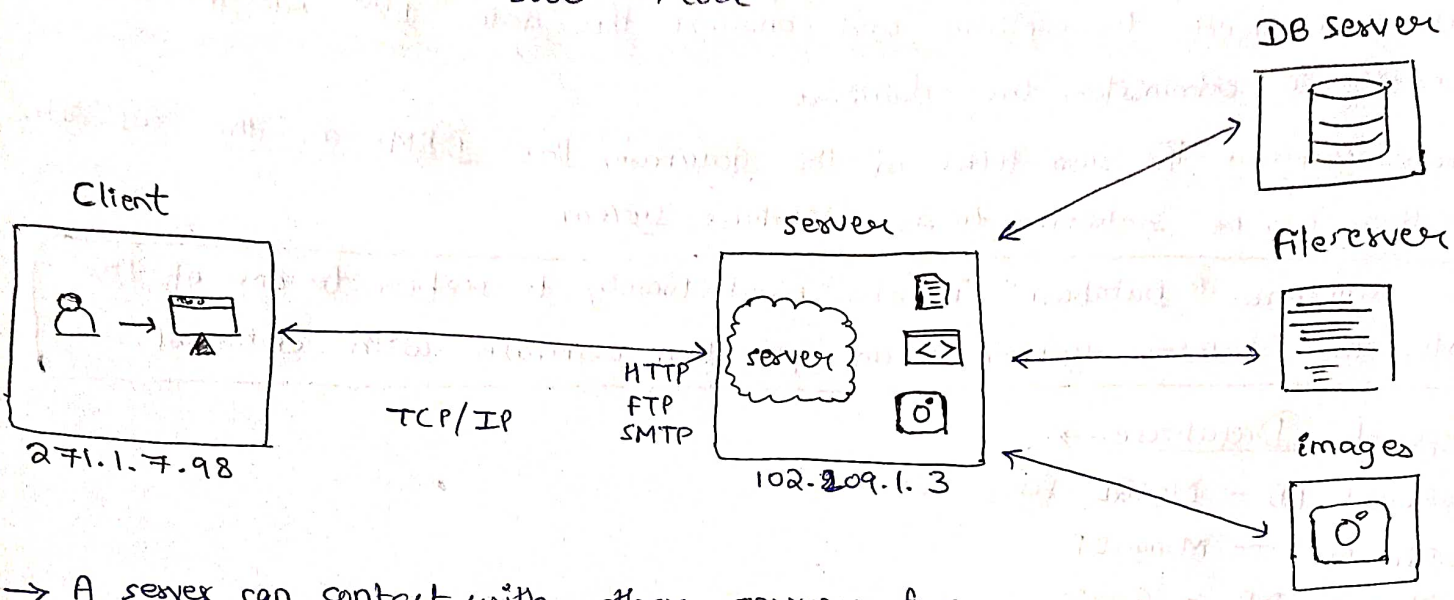


How does server know request is coming to which Application/server? → PORT

PORT → 4 digit → so that user can connect to that server only in a computer, port is to identify the Application server.

Domain Name
⇅
IP + PORT → PATH
⇅
API

example: namastedev.com /api/getuserinfo
123.4.5.6 :3000 → code



→ A server can contact with other servers too.
→ The database, files, images can be kept on different servers and can be Accessed by our server when needed while it can also access data from different servers too.

Creating A Server [using http]

```
const http = require(http)
const server = http.createServer((req, res) => {
    res.end("Hello World")
})
server.listen(7777)
```

- http is low-level, a tough and tedious way, difficult way to handle routes
- Express is preferred because it provides a higher Level of Abstraction than the native http module, simplifying routing, middleware handling and request/response management.
- Express enables the developers to handle ~~manual~~ multiple routing easily, whereas http require manual handling of URL paths.