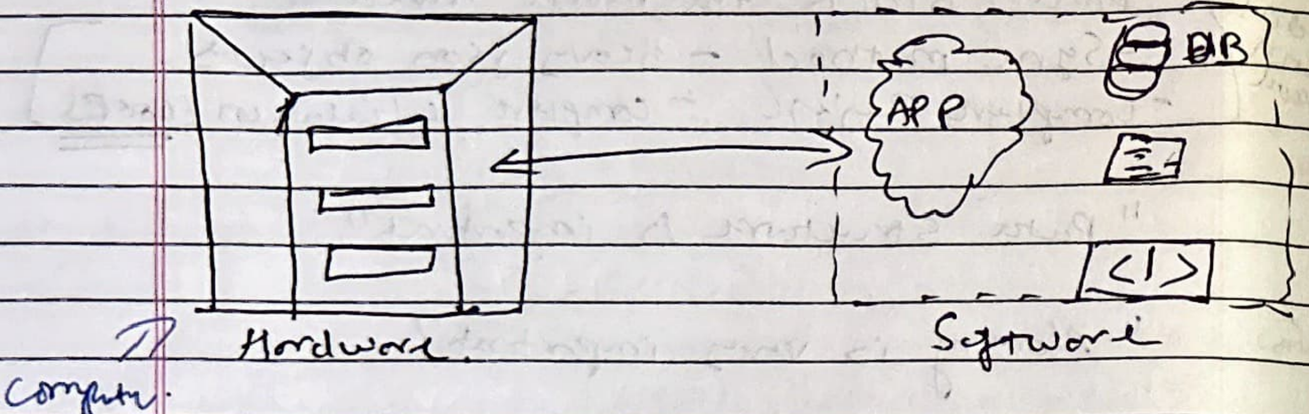
Server.

How web works.



Server can be both, hardware and Software.

Hardware | now we want to outside world to connect
files, data, img | this and can request this file
↑ ↑ ↑
request

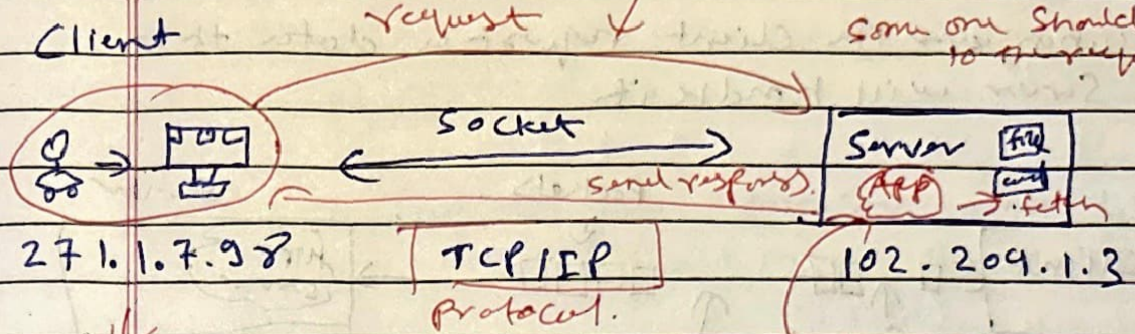
we need a Software to handle a request.

we need application on our Server to handle request.

[ec2 instance] ← we have taken server.

HTTP Server using node.js

then Socket connection is closed.



Some one should be listing to the request

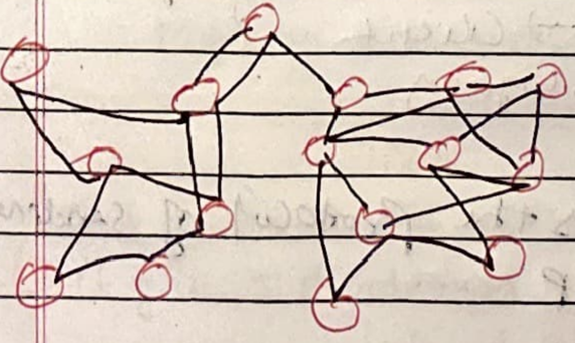
Client becouse it is requsting a data or we can call it a Browser

this is also known as ~~Srvr~~ Server sometime.

There can be multiple Client

TCP/IP ← Basically a rules.

why we call it web.
Suppose we have lots of Client with ip (i) connected to each other with IP



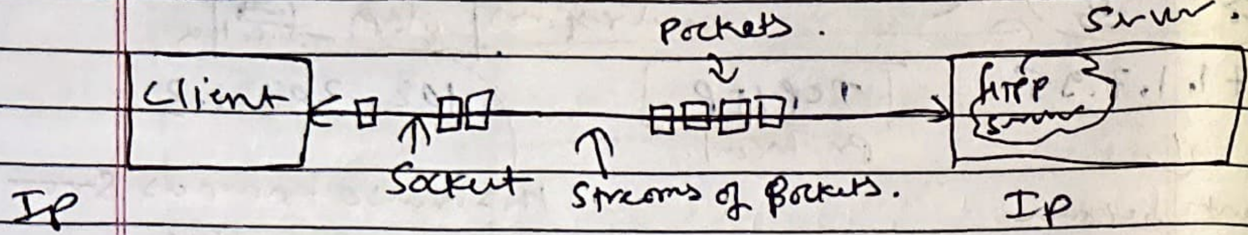
(have format or language) think about.

HTTP - Server
FTP - Server
SMTP-Server
Proxy - Server.

~~[illegible]~~ each have certain rules.
rules and protocols how data will be send.

Hyper text transfer protocol
file transfer protocol
~~sm~~ Simple mail transfer protocol

when ever the client request a data then Server will handle it



TCP/IP .

(it's comes in packets) Stream ~~means~~ means connection
Buffer means chunks we sends.

According to {TCP/IP} Protocol.

[Language that the client and server speaks]
is HTTP, FTP, SMTP
this are languages.

Server ← Talk → Client
using this language

and
{TCP/IP} Protocol is the Protocol of sending the data over on ip

when data is transpreed it is sends in packets this is controlled by [TCP/IP]
Transmission control Protocol
and it is done in packet.

inside node JS we have concept of [Stream and buffer] ← Read it

[Stream, buffer, event emmiter, event listner.]
↳ Read more about it's internal of node

we dont generally comminute via IP,
at the end of the day everything comminucate with IP

[IP - as a longitude and latitude.]
↳ we as humans dont remember

when we make a call to any website eg
google.com, before making a call to google.com
internally it make a call to [DNS servers]

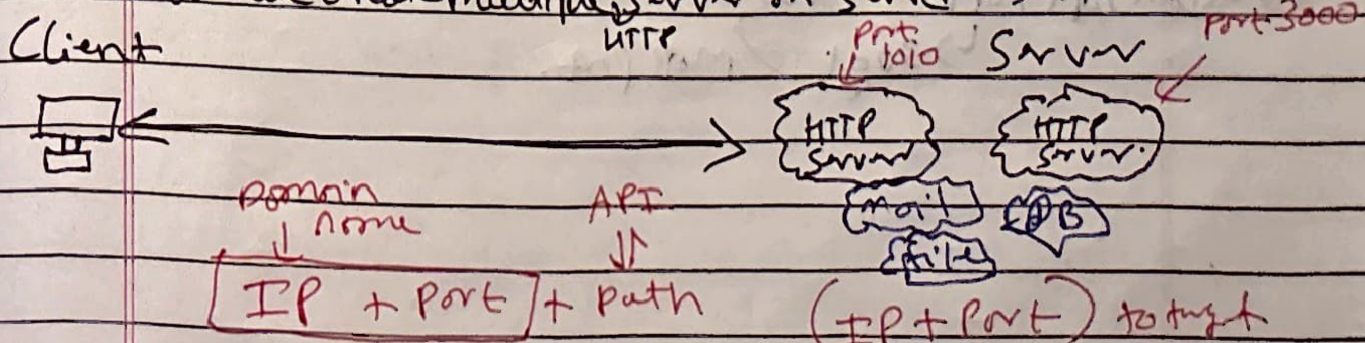
Mapping between ~~DNS~~ Domain name ⟶ IP

DNS

[DNS Reads website url, read the url and then it give IP address]

behind the Scene

Can we create multiple server on server → Yes



How can we know which server to make a request if we have multiple HTTP Server - using [Port Number]

Port no is needed that can other comminute

Domain Name
⇅
IP + Port + Path
↓
API

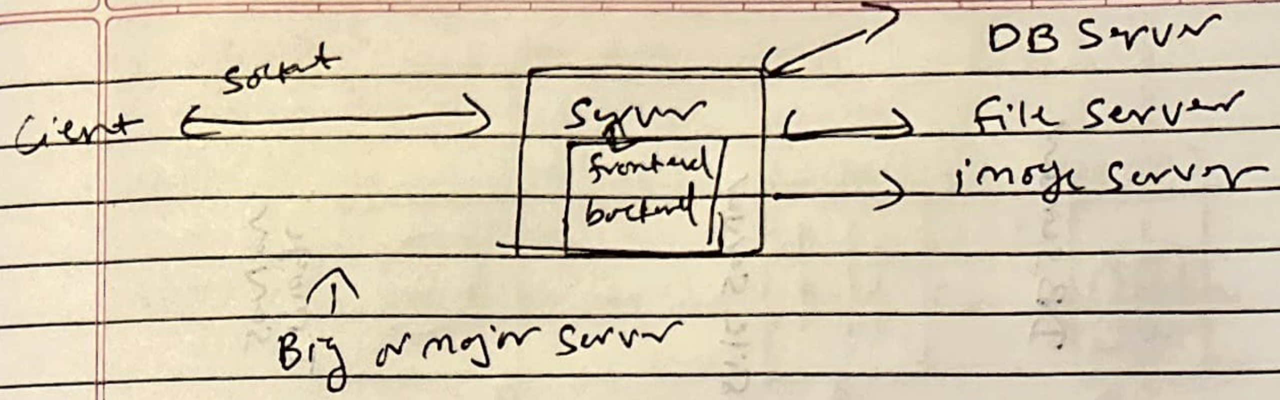
google.com
↑
102.123.1.3.6.7 : 3000 / api / getuserinfo
↓
HTTP Server
Path.

Domain name
(IP) google.com / api / getuserdata
123.4.5.6 : 3000
↑
There can be default port number.

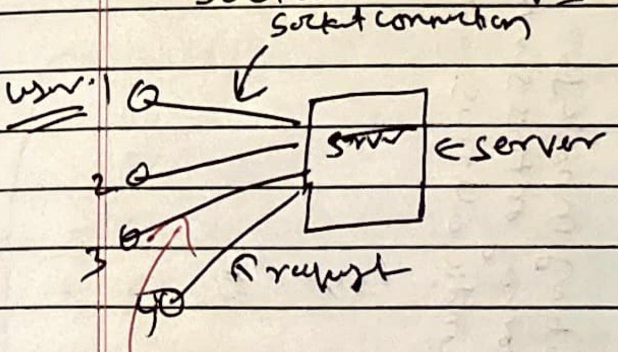
We can map api
↓
3000

Names.com → React
3000
~ mapping Port

Names.com / api / app (data) Node
↳ 3001 map port.



Socket vs web sockets.



Socket connection maked → data request and data respond and then socket closed

open socket ⇔ data get back ⇔ closing socket

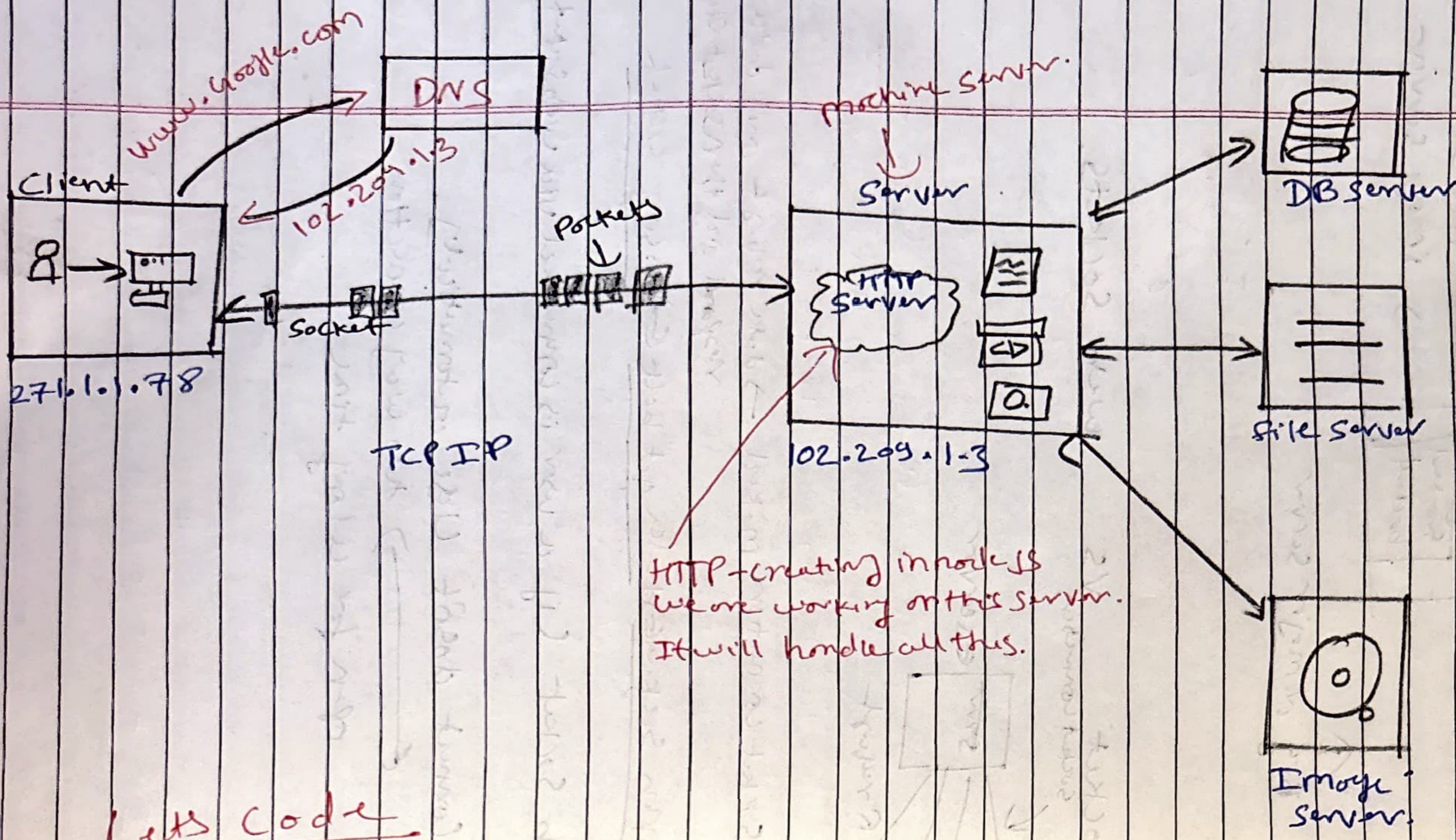
Web Socket (if a user is connected via web socket)

it connect dosn't close automaticaly.

⟷ two way connection

open for a long time.

whole Picture



HTTP-creating in nodeJS
We are working on this Server.
It will handle all this.

Lets Code

Creating our first Server.

Node js have a module called as (HTTP)

HTTP module give us a option Create Server.

Code

```
    const http = require ("http");
- const http = require("http");   "node: http" ← we can also use this.
```

↳ Because it is a native module

```
http.createServer();
```

← this method gives us a instance of a server.

```
- Const server = http.createServer();
- server.listen (7777);
```

↳ now server is ready to Listen.

How to handle a logic

```
const http = require ("http");
const server = http.createServer(function (req, res){

      res.end ("Helloworld'');
```

↳ Sending the data / this method will close or send reply

```
server.listen (7777);
```

```
if (req.url === "/getSecretData") {
        res.end ("There is no secret data");
      }
    res
```

← This is not the best way we use wrapper and use Express js.