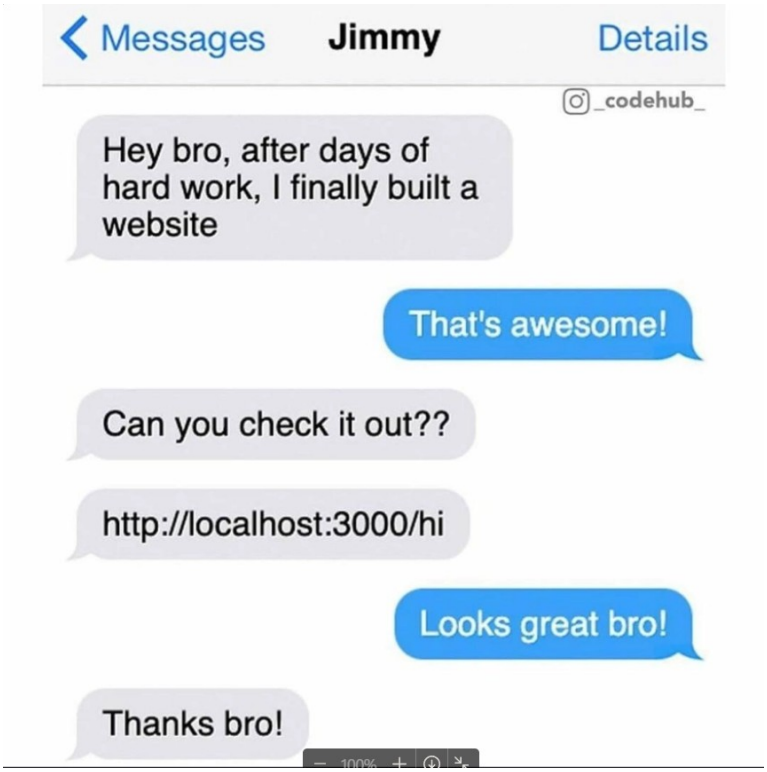
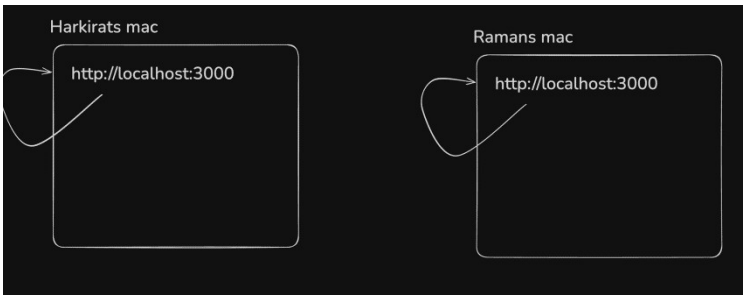



## Domains vs IPs

localhost

"Localhost" refers to the computer you're currently working on. It's essentially a loopback address that points to the machine itself, allowing it to communicate with itself over a network. In technical terms, the IP address for localhost is usually `127.0.0.1` for IPv4, or `::1` for IPv6.

**The localhost simply means a server which is running but loopsback to your own system and we can also do like localhost:3000 but also like 127.0.0.1:3000**




IP is of two types:

**1. IPV4:** in which the digits like 10.10.10.10 each number before decimal must be within 0-255 to be called as a IPV4 and it follows 8 bit code eg 0.0.0.0, 255.255.255.255 each is 8 bit in size , and there are limited number of ip address which can exist in the world isme 2^32 tha

2. **IPV6: isme** 2^128 hai

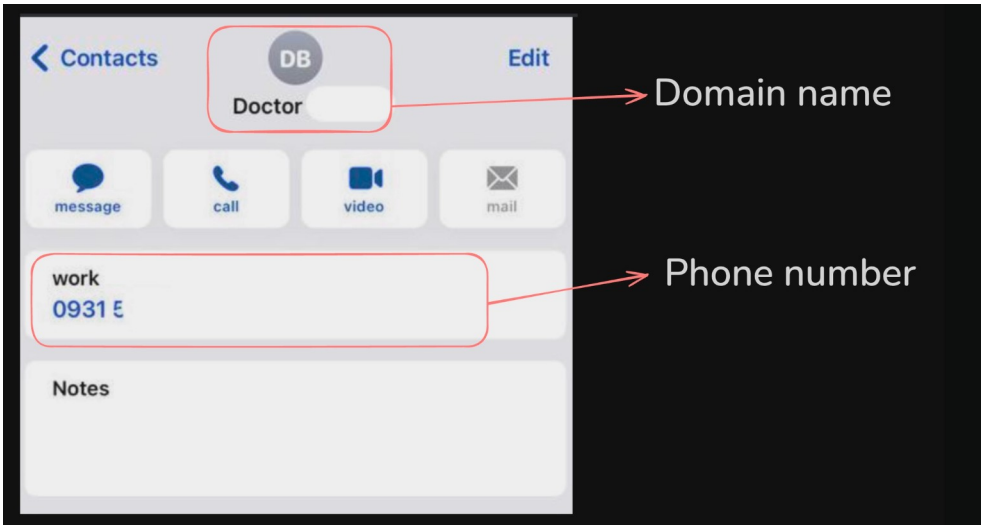
### IP (Internet Protocol) Address:

- An **IP address** is a unique string of numbers that identifies a device on a network, like the internet. Think of it like a **house address**: it's how computers (or any devices on the network) know where to send information.
- For example, `192.168.1.1` is an IP address.
- An **IP address** is essential for routing data on the internet, but it's not the most human-friendly system.

## Domain name

- A **domain name** is the readable, human-friendly address we use to access websites, like `google.com` or `example.org`.
- Domains are a higher-level abstraction that makes it easier for us to remember websites instead of trying to recall a string of numbers (like an IP address).
- For instance, when you type `www.google.com` into your browser, your computer looks up that domain name and finds the corresponding IP address, then connects to the website.

Domain name vs Phone number




## Limited IP addresses

There are limited number of IP addresses in the world (ipv4 specially). So it’s not very easy for us to get a public IP. Most IPs are blocked by cloud providers or Big companies (JIO)

### STUN Protocol

It is a protocol that helps devices behind a NAT (Network Address Translation) router discover:

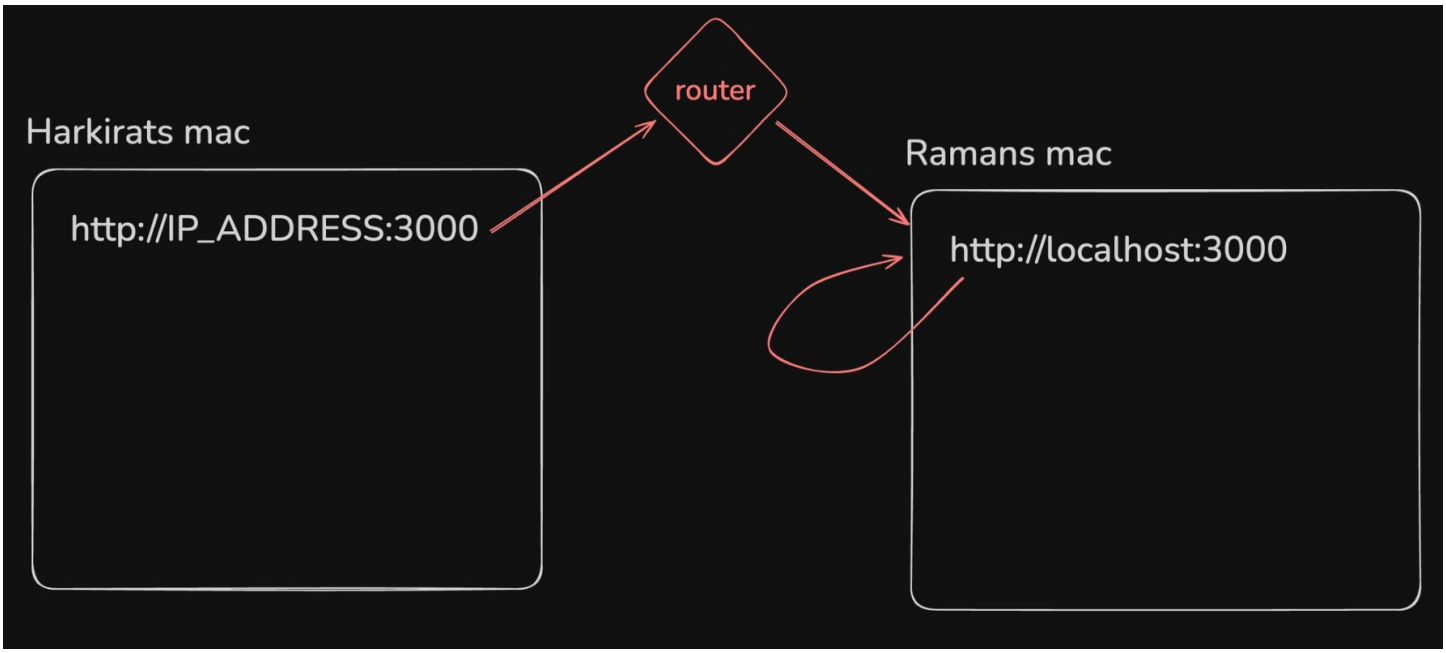
✅ Their public IP address

✅ The type of NAT they are behind

✅ The public port number assigned by the router

# Local network, routing (mild hosting)

If you have multiple laptops on the same `wifi router`, you can access one machine from another by using their private IP address. This is a `mild` version of deploying your app on your `local network` (or whats called the intranet)




## Steps to follow

1. Start a node.js process locally on port 3000

```
const express = require('express');
const app = express();
const port = 3000;

app.get('/', (req, res) => {
  res.send('Hello, World!');
});

app.listen(port, () => {
  console.log(`Server is running on <http://localhost>:${port}`);
});
```

1. Find the IP of your machine on the local network

```
ifconfig or ipconfig
```

our you can also use npx serve

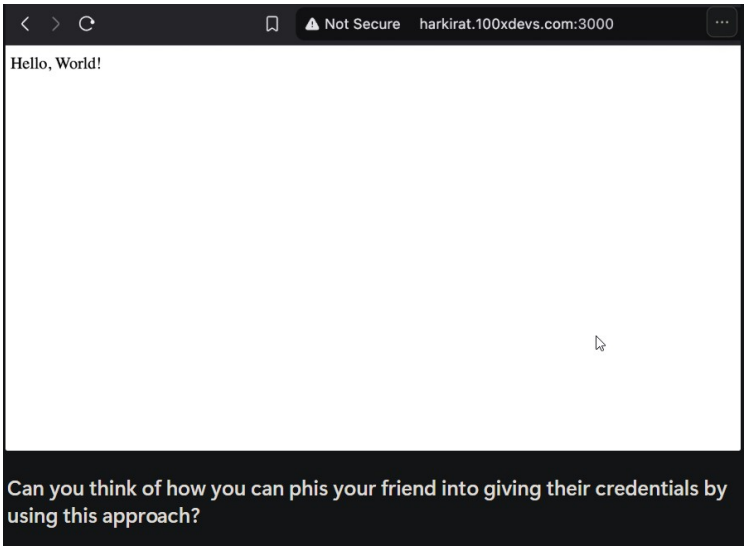
**YOU CAN DO A ROUTER SCAN TO GET ALL THE SYSTEM IPS AND THEN GET INTO IT PORT SCAN**

## HOSTS FILE IMORTANCE

You can override what your domain name resolves to by overriding the hosts file.

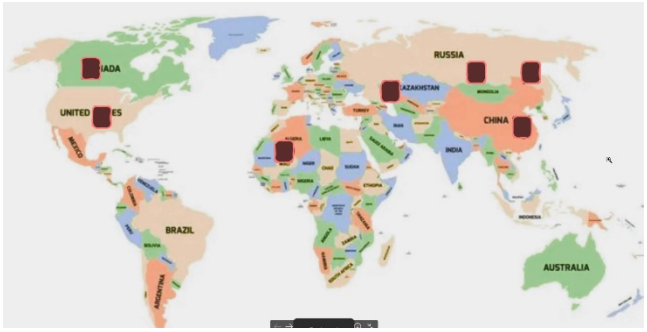
```
vi /etc/hosts
127.0.0.01      harkirat.100xdevs.com
```

you can redirect any website on your machine and then make a same ui and can phish anyone




### Baremetal to host postgress servers

What is a VM

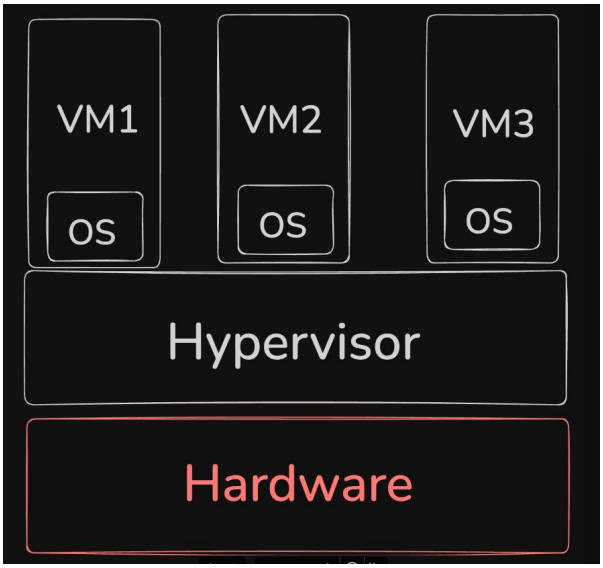



VMs run on a physical server (called the **host**) but are abstracted through a layer of virtualization software called a **hypervisor** (e.g., VMware, KVM). This hypervisor divides the host machine's resources (CPU, memory, storage) into separate virtual machines.

Each VM acts like a completely independent machine, even though they share the underlying hardware. You can run different operating systems and applications in different VMs on the same physical server.

VMs are highly flexible and easy to scale. You can quickly spin up, modify, or delete VMs, and you can consolidate multiple workloads on a single server.

The virtualization layer introduces a slight overhead in terms of performance because the hypervisor needs to manage resources and ensure each VM operates independently. However, with modern hypervisors and powerful hardware, this overhead is minimal.

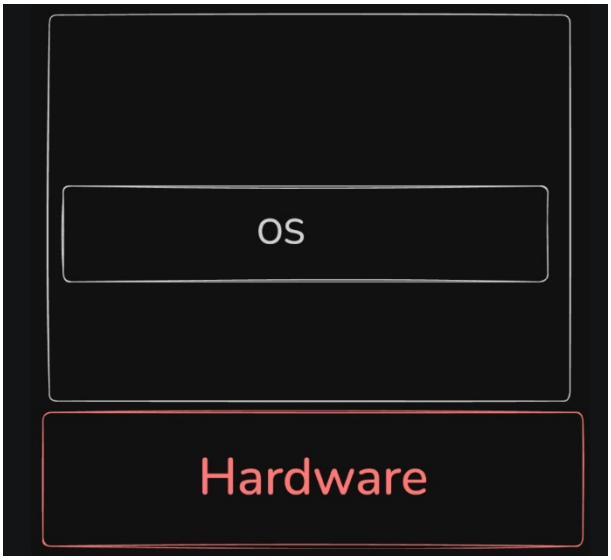



## Bare metal servers(BEST FOR PERFORMANCE)

n a bare-metal setup, an operating system (OS) runs directly on the physical hardware without a hypervisor in between. There’s no virtualization layer.

Since there's no hypervisor, bare-metal systems tend to offer better performance, as the OS can directly access all the server’s resources without sharing them with other instances. This is especially important for high-performance applications like large databases, gaming servers, or mining crypto

With bare-metal, you’re typically limited to the resources (CPU, memory, storage) of the actual physical server. You can't dynamically allocate resources like you can in a VM.




**Some servers options:**

AWS, GCP, Azure

Easy  one: Civo, DigitalOcean, Vultr**SSH protocol, password based auth**

The **SSH protocol** (Secure Shell) is a cryptographic network protocol that allows secure communication between two systems, typically for remote administration. It's most commonly used to log into remote servers and execute commands, but it also facilitates secure file transfers and other operations.

**Key Features of SSH:**

1. **Encryption**: SSH encrypts the data that's sent between the client and the server, so even if someone intercepts the connection, they can't read the data. This makes it much more secure than older protocols like Telnet or FTP, which transmit data in plaintext.
2. **Authentication**: SSH can use two methods of authentication:
    - **Password-based**: You enter a password to authenticate yourself to the remote system.
    - **Public Key-based**: A more secure method, where the client uses a private key to authenticate, and the server checks it against the corresponding public key. This eliminates the need for passwords and provides an extra layer of security.
3. **Integrity**: SSH ensures the integrity of data, meaning that data cannot be tampered with while it's in transit. If someone tries to alter the data being sent, the connection will be immediately disrupted.

**Password based**

While setting up a server, select password based authentication

**Example from digitalocean**

Command to use

ssh ubuntu@SERVER_IP

or

ssh root@SERVER_IP

if the system is saying in windows that not enough permissions

then first disable inheritance and then remove the other users from the security tab and also give more permissions to the user such as read , read and write, read and execute

done the problem is fixed

All cloud providers have a different name for their machine

On Digitalocean its called droplets, on aws its called ec2(elastic compute v2)

These website gives two methods to access the machine which is:

1. Passwords based
2. Public key based

The password based approach is easy but there is a problem which is if I have given like access to multiple users and someone does something wrong you wont be able to know that, you may say we can trace there  ip but maybe assume he was using a VPN too. So to fix that we use this:

**SSH protocol, ssh keypair based (using ssh keys it's a better idea to connect to these vms)**

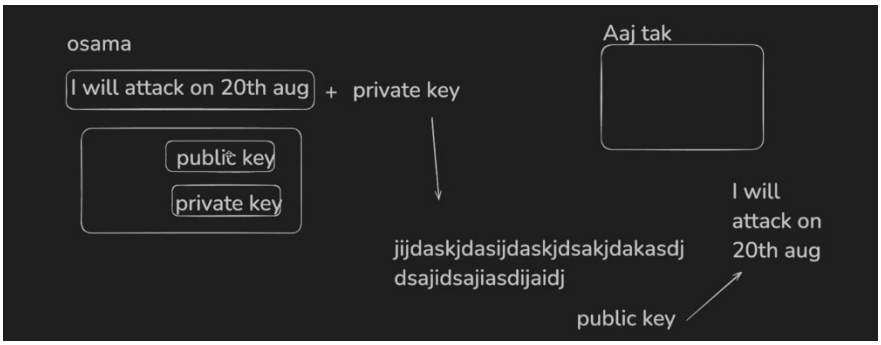
on github also there are two methods to download the repo using ssh or https

things used to create private keys are:

hashing, encryption, public key cryptography, RSA, ed25519, edcdsa

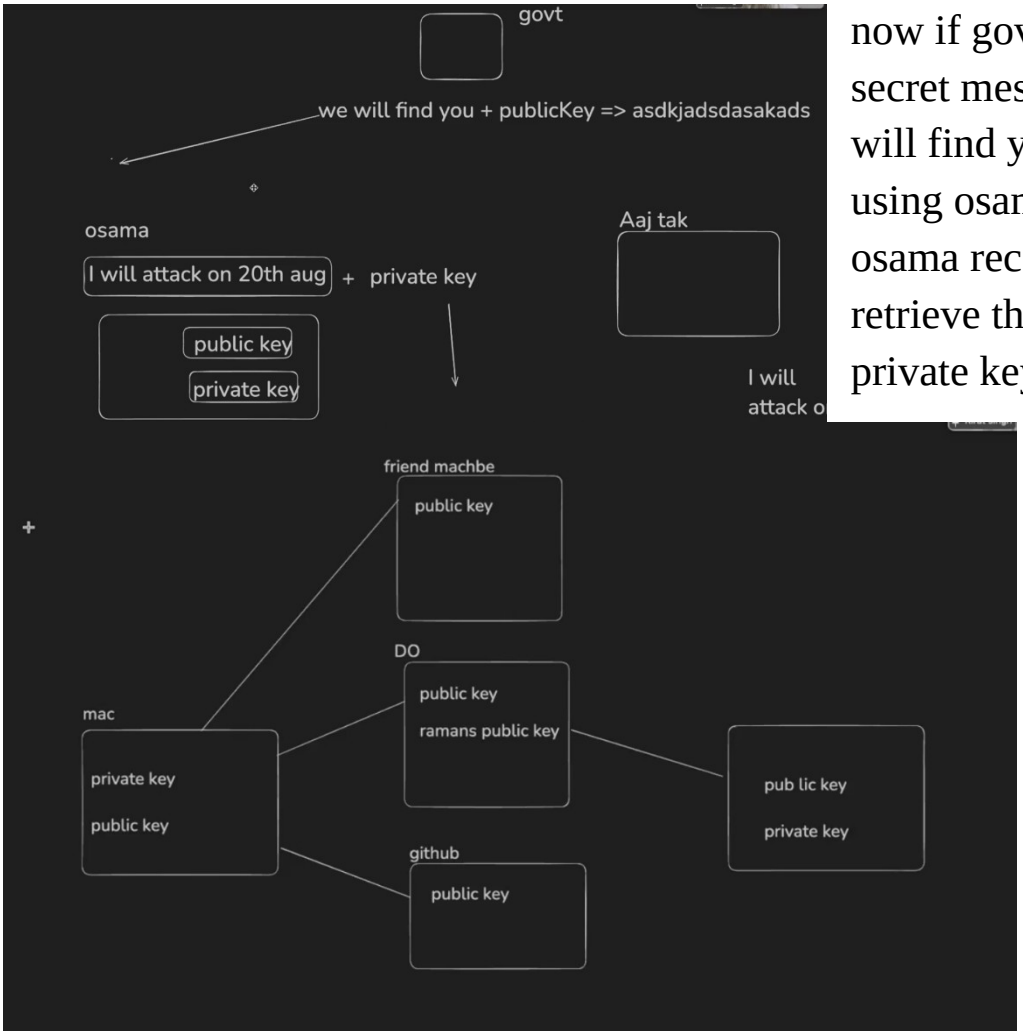
**PUBLIC PRIVATE KEY IDEA**

**USECASE 1**




imagine osama want to broadcast a message, so what he will do first he will send his local public key to everyone and then he will sign the message with his private key which will return something random gibberish and that gibberish can only be decoded using his public key and then they can identify the message and also this confirms that the msg was sent from osama only

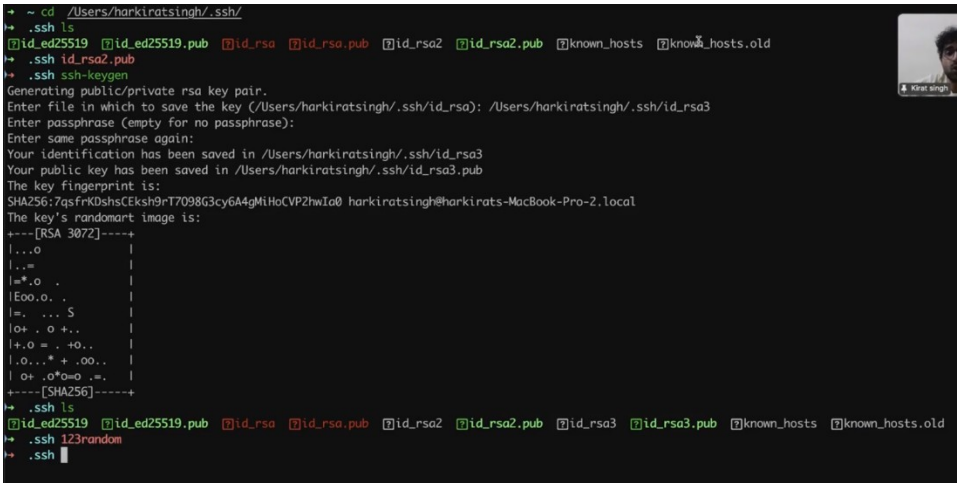
**USECASE 2**




now if govt want to send some secret message to osama like we will find you they have to sign it using osama publicKey and then osama receives that only he can retrieve the text they sent with his private key no one else can't

to connect to DO(digitalocean) we have to first add a ramans public key in the DO to give access to raman and when raman tries to connect he can connect and DO will check his private key if it matches it will simply let him connect

and if he does something bad we will get to know that




**while creating a public private key we can also add a top level password to secure it more if someone copies that file then he has to enter that password also to access the keys**

**how to create a public private key show above?**

1. ssh-keygen
2. choose a location and then set up a password

Explore your public and private key

for windows **C://users//.ssh//authorized_keys**

**the pub is the public key and the normal one is the private key**

cat ~/.ssh/id_rsa.pub

cat ~/.ssh/id_rsa

- Try adding it to Github so you can push to github without password
- Try adding it to digitalocean and ssh using it.

ssh ubuntu@IP

or

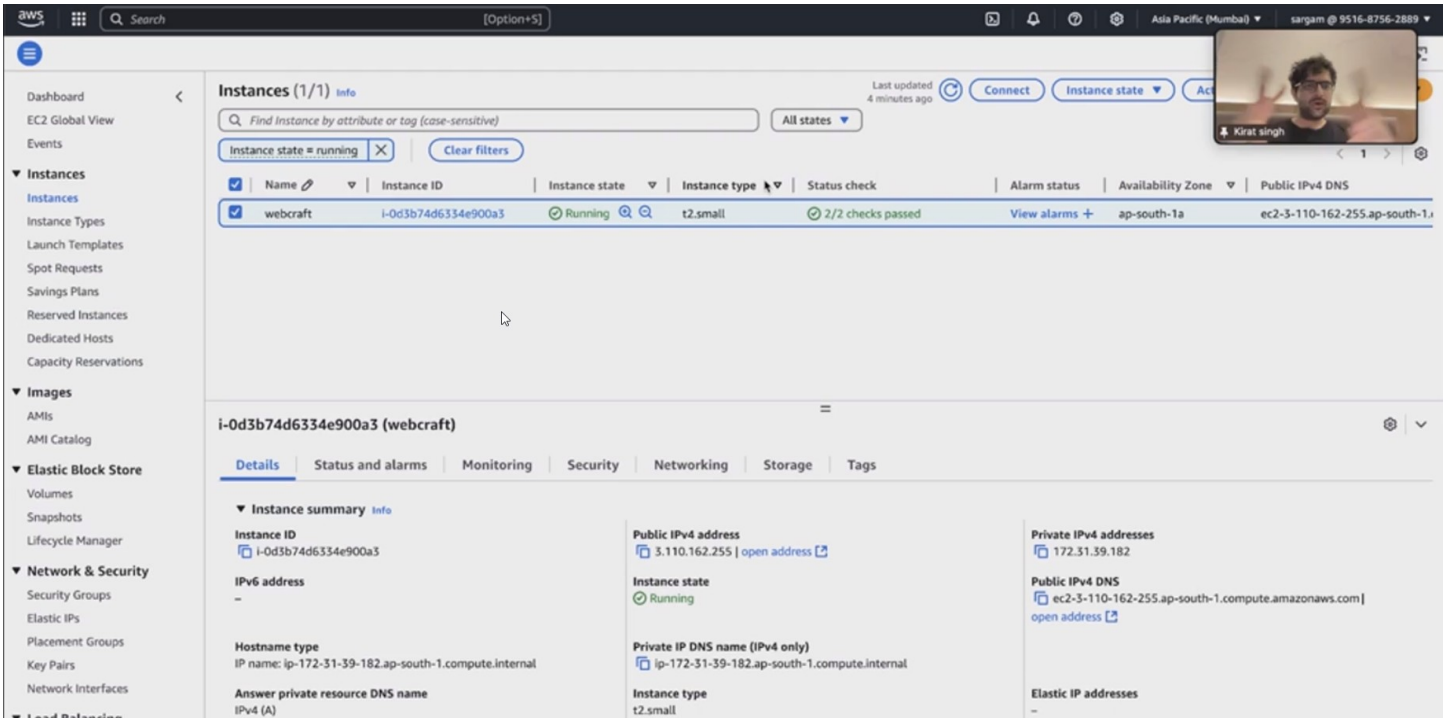
git clone git@github.com:100xdevs-cohort-3/week-24-deposit-with-infra.git (try a private repo)


**Check authorized_keys**

NOW IN GITHUB YOU CAN ADD THE PUBLIC KEY AND THEN YOU CAN CLONE THE REPO USING SSH ALSO AND ALSO IN DO YOU CAN CHOOSE SSH KEY OPTION AND THEN ADD THE SSH PUBLIC KEY AND THEN WHEN YOU TRY TO ACCESS THAT MACHINE YOU NEED TO PUT YOUR PASSWORD PHRASE AND THEN YOU CAN USE THE MACHINE AND INSIDE THE VM MACHINE THERE IS JUST A FOLDER NAME **AUTHORIZED_KEYS** WHICH CONTAIN YOUR PUBLIC KEY AND THAT'S WHY IT LET YOU ENTER AND FOR ANYONE ELSE YOU WANT TO ADD YOU CAN JUST PASTE HIS IP

SETTING UP AWS

elastic net = a ip assigned which will remain same even when we restart the system



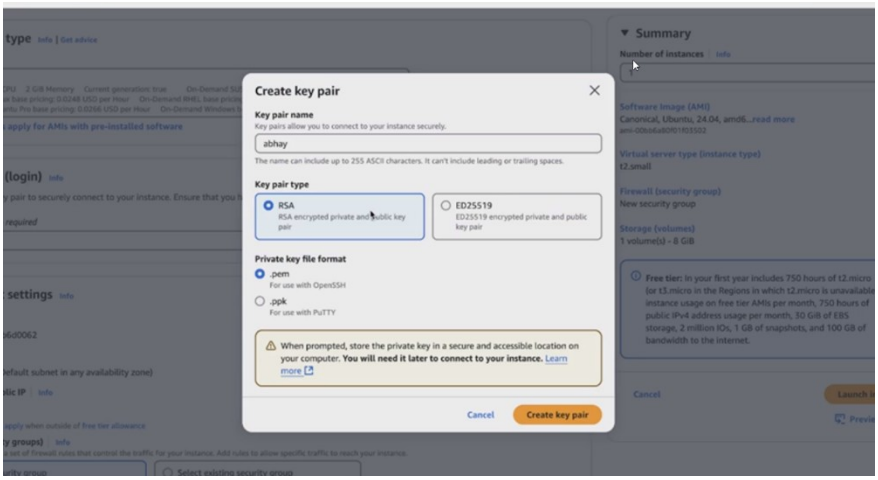

To setup first go to login and then

Search for ec2

Then launch instance

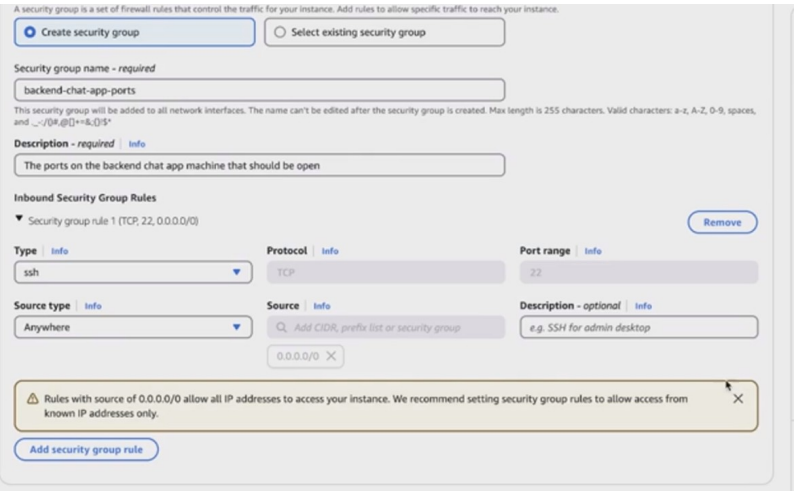
**SOME  JARGONS C GROUP , JUDGE 0**

T2small and t2 micro has a issue that when I will do npm run dev it can go out of memory issue but you can do npm serve




this helps to give access to new member and also the rsa and ed is a algo which helps to make a public and private key.

In do and vulter there is a **firewall** and in aws there is a **security groups**




c group = its like limiting resources to the user based on there usage

type is like when we want to use express, http we use like ssh,

port range this we use when we like require to run multiple backend and frontend

Source type = kaha sae request allowed hona chaiye

SUBNET = bunch of ip ranges used in source type

Security group help us do that thing by securing the ports and all

**Different instance types have different usecase like some may be better for video transcoding other maybe better for other reason**

NETWORK SETTINGS SECURITY GROUP IS VERY IMPORTANT

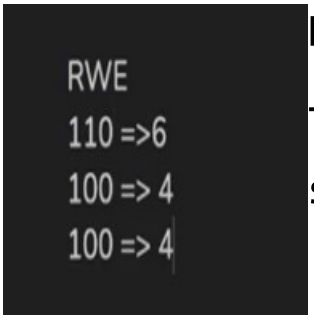
to to use the

ssh ubuntu@ip

chmod 400 pdkey.pem (DO IN GITBASH)

ssh -i pdkey.pem ubuntu@16.170.98.95

ssh -i abhay.pem ubuntu@ip




RWE = read , write, execute

first person has read and write access

second and third only read

chmod 700 kirat-class.pem

in this the 700 means the owner should have all the perms

so it gives all the access to the file to the user

after this we can run the ssh command

then in the terminal of the system do the

git clone and npm install commands to install the node also then just npm install

and then run the node filename

but the problem is we opened the port 3000 and 22 but we can change that in the security option

now we can use the ip to run the server yuhuu we did it

ec2 machines are mostly used for deploying backend so yaa

nginx ≡ is said like enginex its same like apache, haproxy, traffic,

why do we need this?

We have a ugly url like 44.44.42.21:8080

so to remove this type of url we need to use **NGINX**

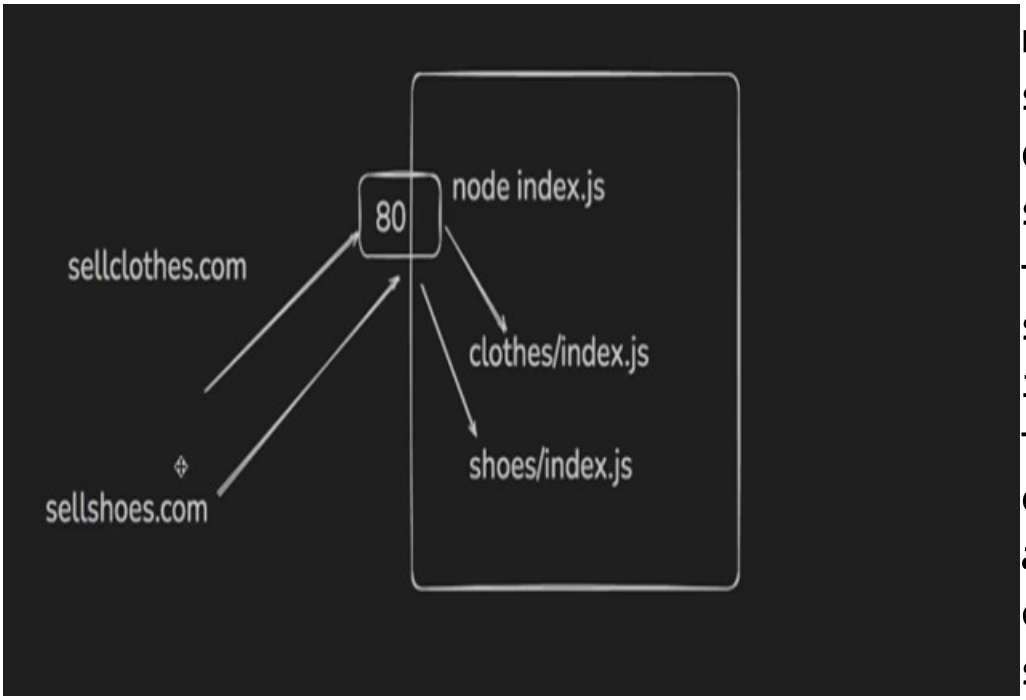
domain the amazon also gives but the :8080 is still there to remove that we use this

if at starting there is https means that is 443 so google has that but the amazon one does not and for that it shows the not secured thingy

for http its 80

so to fix this temp we can run it on port 80

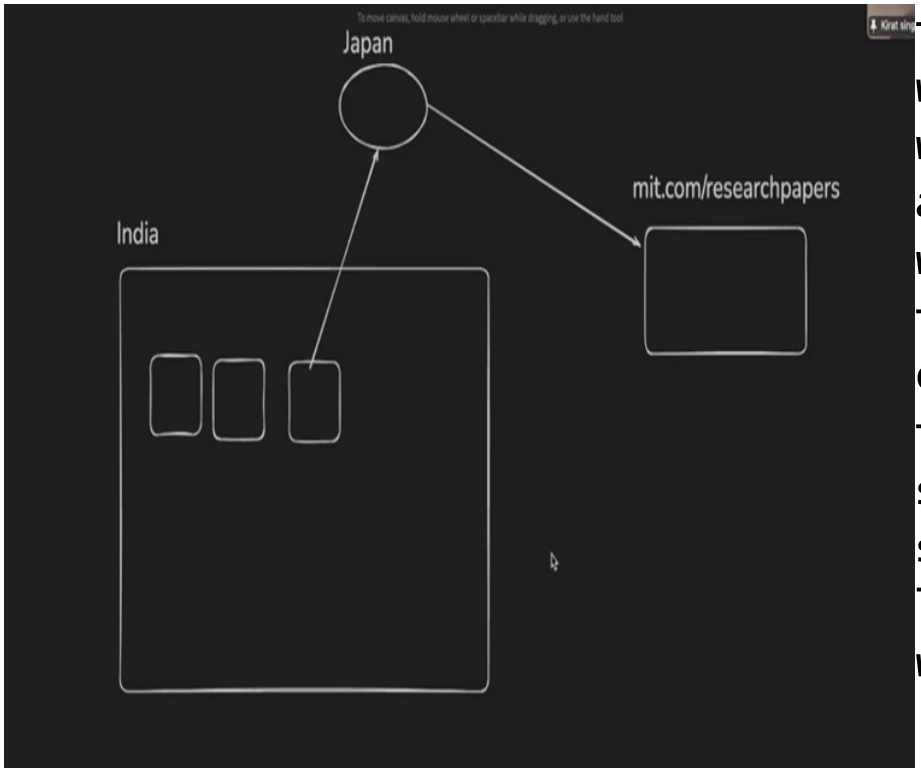
BUT WE CANNOT RUN ON 80




now the issue I s we want that on port 80 if someone comes for sellclothes.com it should take the clothes/index.js and someone comes for sellshoes.com it should go to shoes/index.js

so for that we need nginx

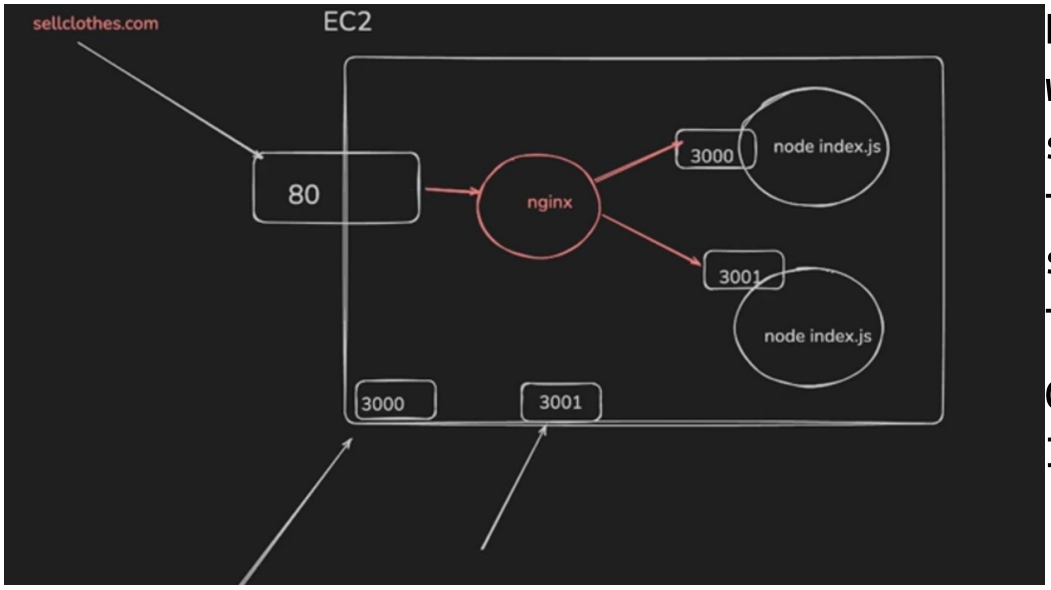
THIS IS CALLED AS REVERSE PROXY




this is how proxy works same like vpn we are trying to access like mit website but imagine they have blocked other users so you try to connect to some authorized servers which will let us connect to the website.

This is called as forward proxy

WHAT IS REVERSE PROXY?




Proxy is like in which we hit the server and then through the server we hit the site

Client → Proxy → Internet Server

Client → Reverse Proxy → Backend Server

but in this when we hit the website the server itself decides that to which port to transfer the request this based on the request is called as reverse proxy

there is something like **load balancer**

**PROXY**

You don’t talk to the website directly.
You talk to a middleman, and the middleman talks to the website.

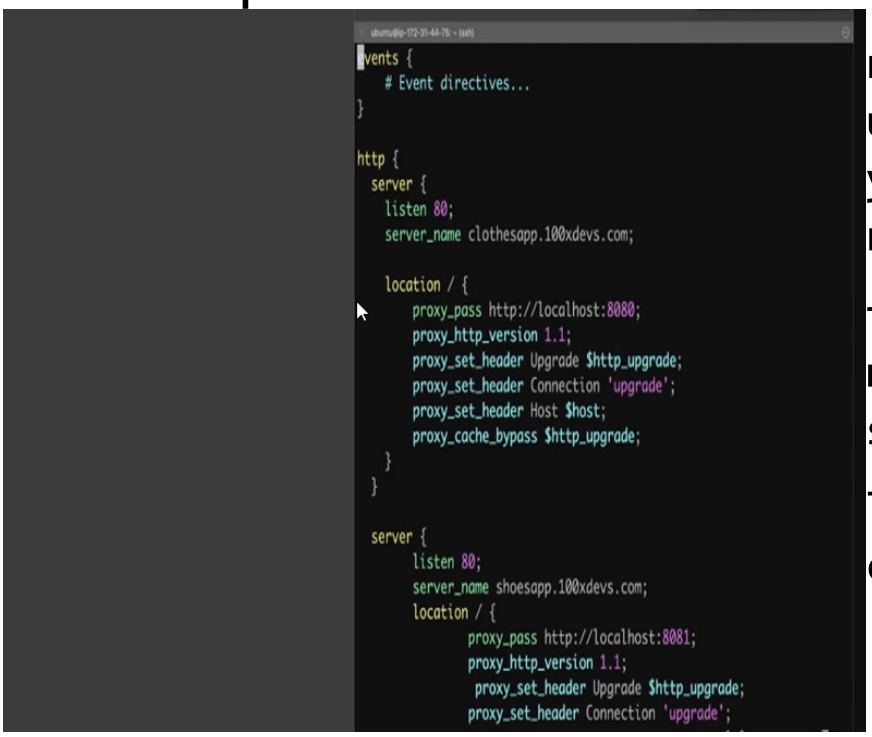
**REVERSE PROXY**

You think you are talking to one website,
but actually there is a middleman sending your request to different servers.

When you add a ip in a record it takes some time to expand its dns across the world and area and then starts to work

you can check it using **ping url**this nginx helps in using same backend for two websites

**sudo == super user do**




now after making the urls routes to the ip you have to setup the nginx config

this is like running multiple projects on same server

like when the req comes on 80 from

clothespp it will redirect to 8080 url and other same also

in future if you want to add more you have to create same file in this like the top one

to edit use sudo vi /etx/nginx/nginx.conf

then do sudo nginx -s reload

80 is for  http

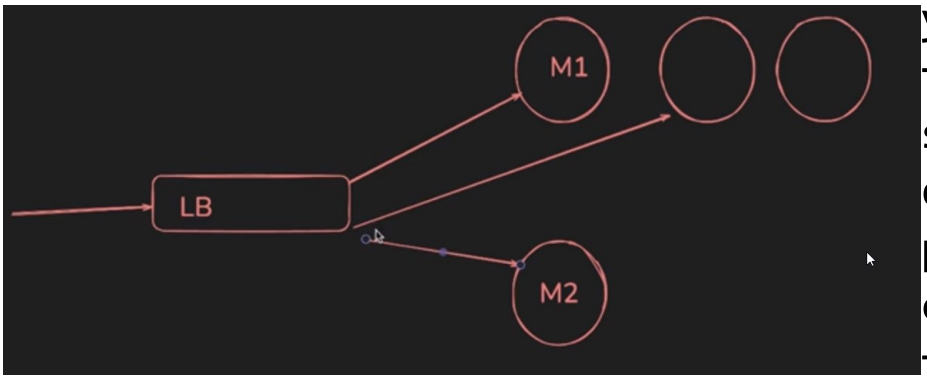
443 is for https

and we don’t need to open the 8080 , 8081 port also

the nginx will do it itself

WHAT IS LOAD BALANCER?

Its like you have to push a update without closing the app what you do is




you send the code in the m2 and then you start to close the older machine and put them with new code below and then transfer the traffic to the new machine and then do this again and again

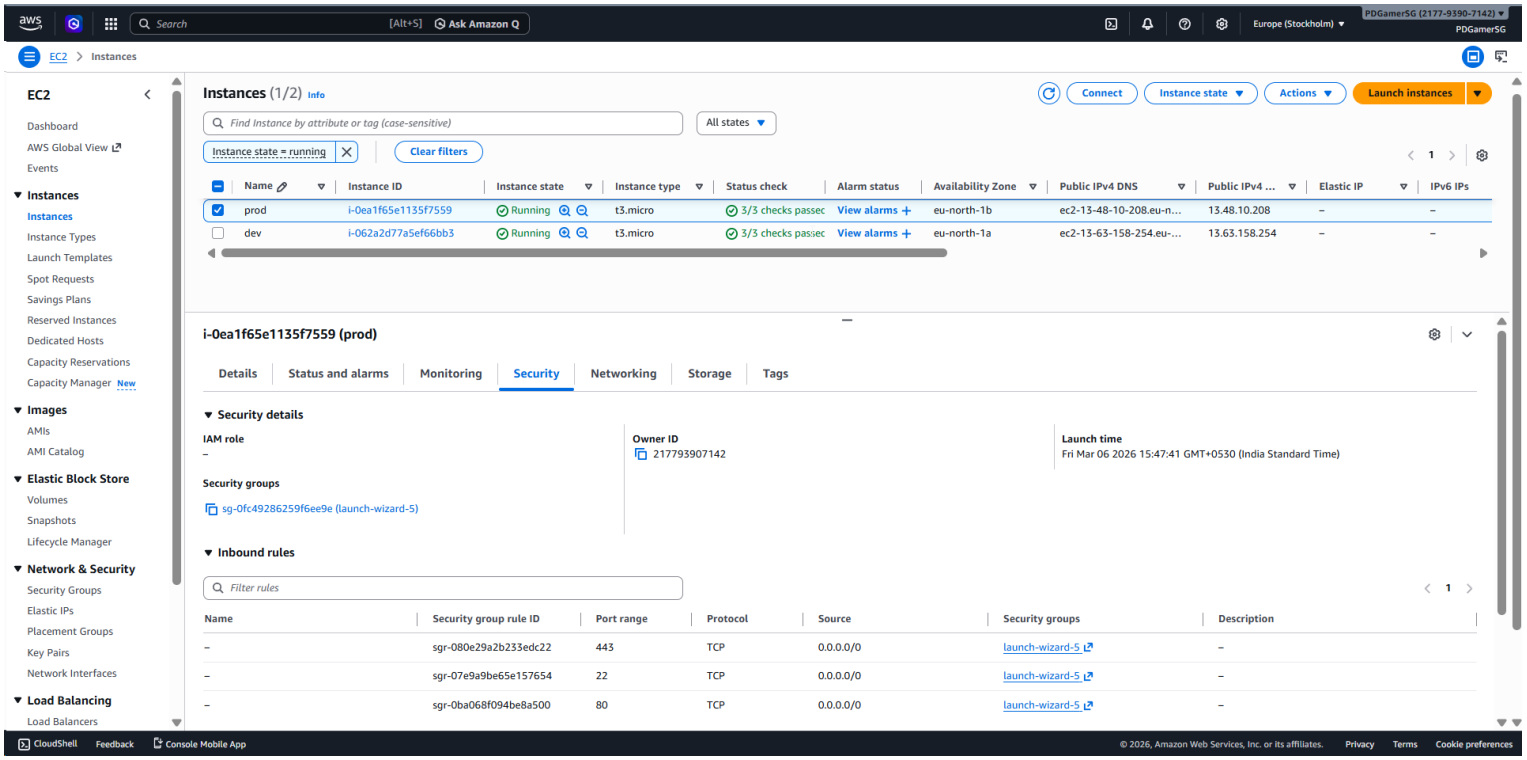
its like down size the old code and upscaling the new code

take code from here for nginx

https://projects.100xdevs.com/tracks/g0AcDSPl74nk45ZZjRdU/aws-7

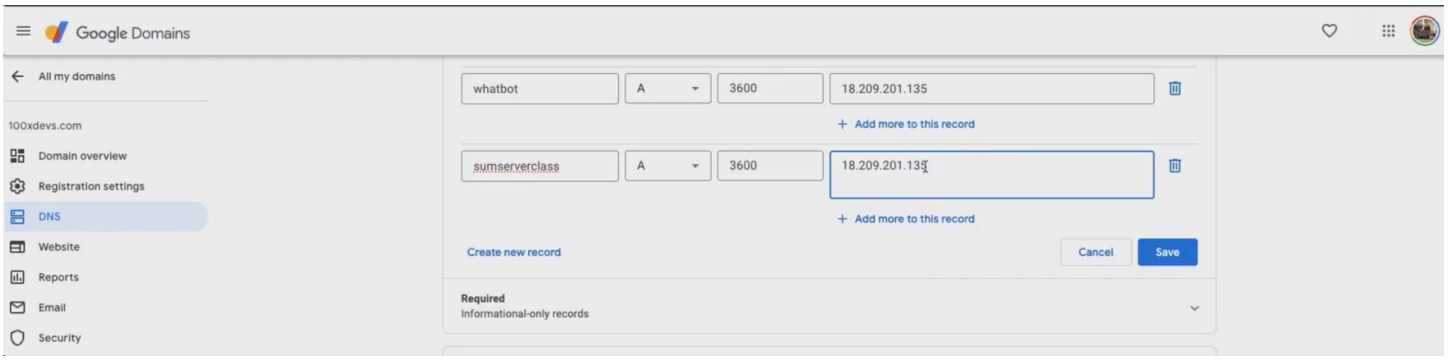
To install node
https://www.digitalocean.com/community/tutorials/how-to-install-node-js-on-ubuntu-20-04

to make this work need to open these ports also




BY CLICKING ON THE LAUNCH WIZARD YOU HAVE TO ADD THE 3000, 3001, 3002 PORT TO MAKE IT WORK ON THE URL OTHERWISE THE MACHINE URL+ THE PORT WILL NOT WORK
you can check dns using
check dns propagation
click  on the security group to open the page and then add the inbound security rules

http://staging.week-25-http.pallabdas.me:3000,3001,3002



```
ubuntu@ip-172-31-92-166:~/test/sum-server$ sudo vi /etc/nginx/nginx.conf
ubuntu@ip-172-31-92-166:~/test/sum-server$ sudo certbot --nginx
Saving debug log to /var/log/letsencrypt/letsencrypt.log

Which names would you like to activate HTTPS for?
We recommend selecting either all domains, or all domains in a VirtualHost/server block.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1: admin.100xdevs.com
2: app.100xdevs.com
3: app3.100xdevs.com
4: bounty.100xdevs.com
5: bounty-admin.100xdevs.com
6: db.100xdevs.com
7: projects.100xdevs.com
8: sum-server.100xdevs.com
9: sumserverclass.100xdevs.com
10: userapp.100xdevs.com
11: week-18-class.100xdevs.com
12: whatbot.100xdevs.com
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Select the appropriate numbers separated by commas and/or spaces, or leave input
blank to select all options shown (Enter 'c' to cancel): 9
Requesting a certificate for sumserverclass.100xdevs.com

Successfully received certificate.
Certificate is saved at: /etc/letsencrypt/live/sumserverclass.100xdevs.com/fullchain.pem
Key is saved at:         /etc/letsencrypt/live/sumserverclass.100xdevs.com/privkey.pem
This certificate expires on 2024-07-15.
These files will be updated when the certificate renews.
Certbot has set up a scheduled task to automatically renew this certificate in the background.

Deploying certificate
Successfully deployed certificate for sumserverclass.100xdevs.com to /etc/nginx/nginx.conf
Congratulations! You have successfully enabled HTTPS on https://sumserverclass.100xdevs.com

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
If you like Certbot, please consider supporting our work by:
 * Donating to ISRG / Let's Encrypt:   https://letsencrypt.org/donate
 * Donating to EFF:                    https://eff.org/donate-le
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

## Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

### Inbound rules Info

| Security group rule ID | Type Info | Protocol Info | Port range Info | Source Info | | Description - optional Info | |
|---|---|---|---|---|---|---|---|
| sgr-0b0297e4254eab96c | Custom TCP | TCP | 8080 | Custom | 0.0.0.0/0 | | Delete |
| sgr-0b3b04a70566cdf3f | HTTPS | TCP | 443 | Custom | 0.0.0.0/0 | | Delete |
| sgr-0c8c5e12376ac117d | HTTP | TCP | 80 | Custom | 0.0.0.0/0 | | Delete |
| sgr-06c5bac31c8d6bc56 | SSH | TCP | 22 | Custom | 0.0.0.0/0 | | Delete |

Add rule

⚠ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Cancel Preview changes Save rules