

AWS Deployment

(<https://projects.100xdevs.com/tracks/g0AcDSPI74nk45ZZjRdU/aws-7>)

What is AWS?

Anyone can rent server. It let us create and rent server

AWS is Amazon's cloud service.

It let's you

1. Rent servers
2. Manage domains
3. Upload objects (mp4 files, jpgs, mp3s ...)(Even videos can be stored in the AWS)
4. Autoscale servers
5. Create k8s clusters

...

The offering we will be focussing on today is **Renting servers**

How can we launch an instance over there, machine in data center of AWS and run some code.

EC2 servers

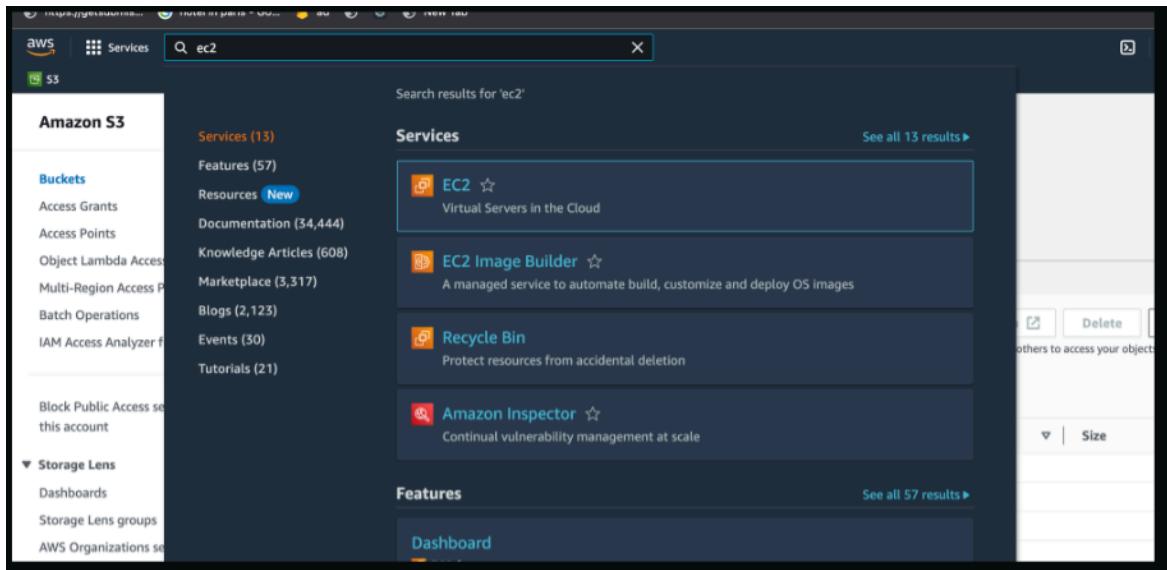
Virtual machines on AWS are called EC2 servers

EC2 stands for elastic compute version 2

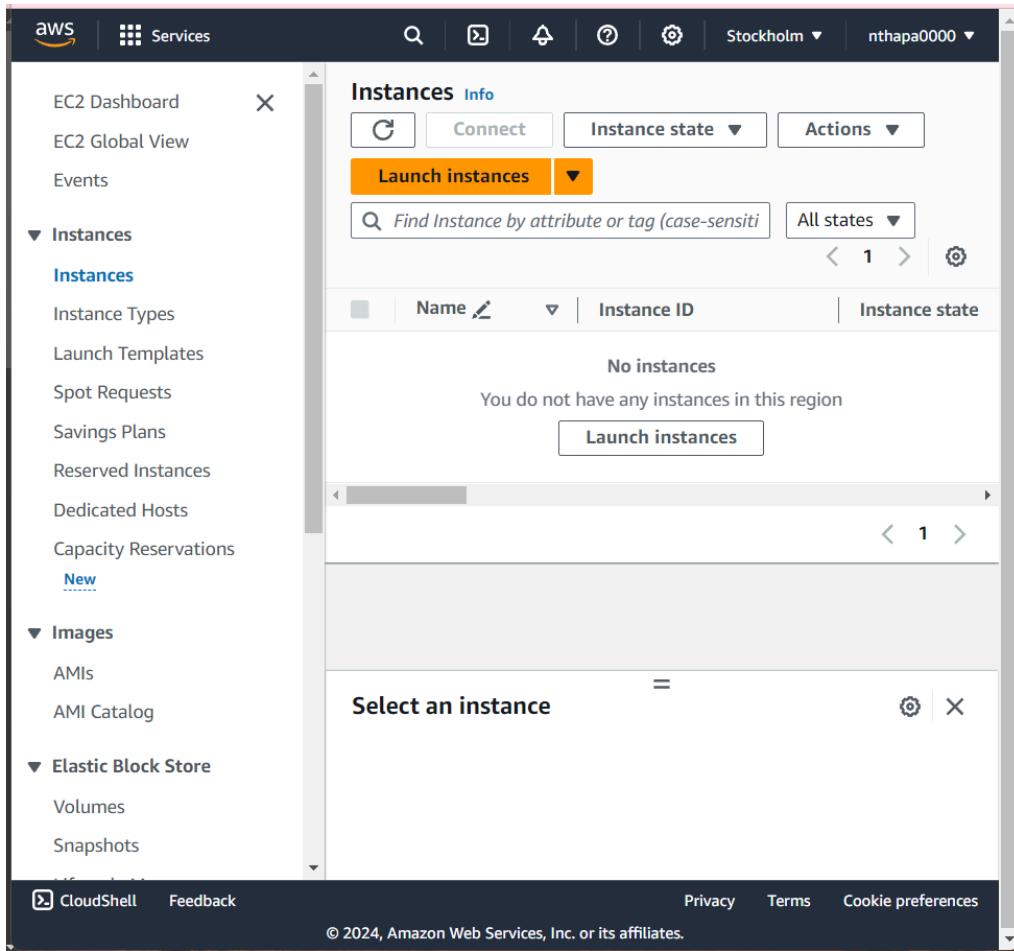
1. Elastic – Can increase/decrease the size of the machine
2. Compute – It is a machine

You can spin up a new EC2 instance from the aws dashboard

It is a place where we can rent a server. We can create multiple EC2 instance

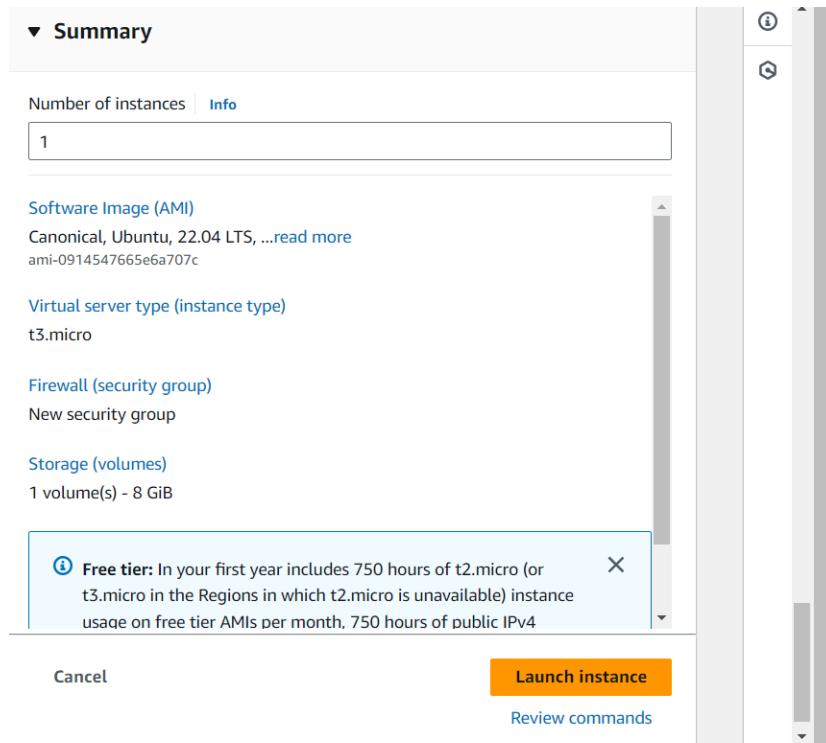


Type Ec2



Click on Launch instances. Fill the details

Creating a new EC2 server



When we launch an instance , server will be created but we still have to connect to it.

Remotely access it and deploy it one easy way to do is **Key pair**

thapa-password.pem downloaded in pc , the name is give by us to key-pair

Cloud providers

What is cloud?? Cloud, we as developer don't care where our code is running, big companien have very big data center where they deploy lot of machine and let people use it to deploy it.

AWS Datacenter a machine has a public IP , hence anyone can visit it. Through routers. And we can use that machine to deploy our code.

▼ Instance type [Info](#) | [Get advice](#)

Instance type

t3.micro	Free tier eligible
Family: t3 2 vCPU 1 GiB Memory	
Current generation: true	
On-Demand RHEL base pricing: 0.0708 USD per Hour	
On-Demand SUSE base pricing: 0.0108 USD per Hour	
On-Demand Linux base pricing: 0.0108 USD per Hour	
On-Demand Windows base pricing: 0.02 USD per Hour	

[Additional costs apply for AMIs with pre-installed software](#)

All generations

[Compare instance types](#)

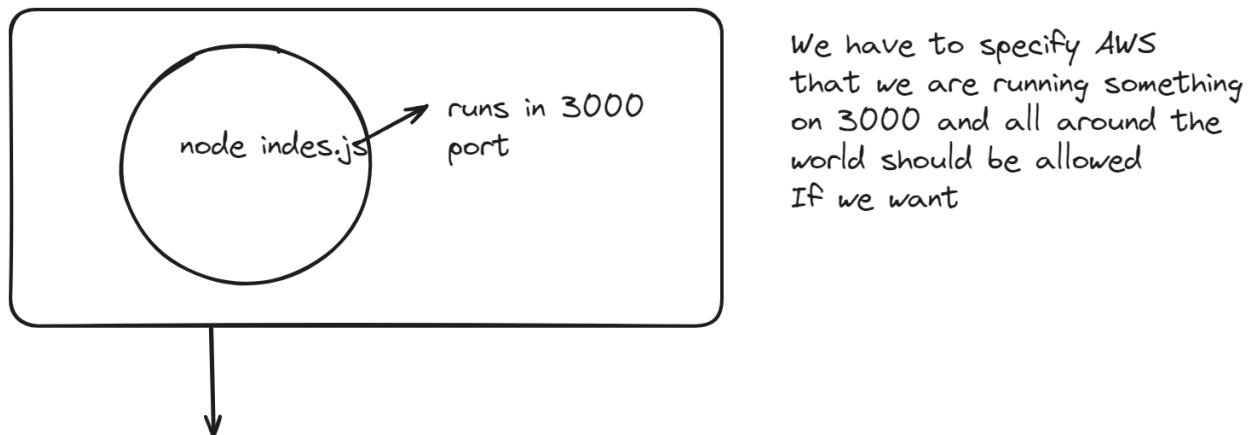
▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

[Create new key pair](#)

They have different instance types where some have lot of no. CPU , where some are optimized for videos etc



we can access this specific server , have some IP(IP of the server (3.1.5.7)) : 3000
apx100xdev.com:3000 if we got some domain

Good practice is that only 443 which is https port is open other ports are rejected by AWS

▼ Network settings [Info](#) [Edit](#)

Network [Info](#)
vpc-06fd008d8f3a3ca7a

Subnet [Info](#)
No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)
Enable
Additional charges apply when outside of [free tier allowance](#)

Firewall (security groups) [Info](#)
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group Select existing security group

We'll create a new security group called 'launch-wizard-1' with the following rules:

Allow SSH traffic from Anywhere
Helps you connect to your instance
0.0.0.0/0

Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

Control what all we have exposed in the Internet

We opened the port 80 and 443 by selecting this IP

If we want to restrict the users, suppose only users from India can open it, or only people with specific IP

Http traffic? Secure Shell (A protocol run in Our machine to run this server)
If we want to connect to server we will open the ssh port

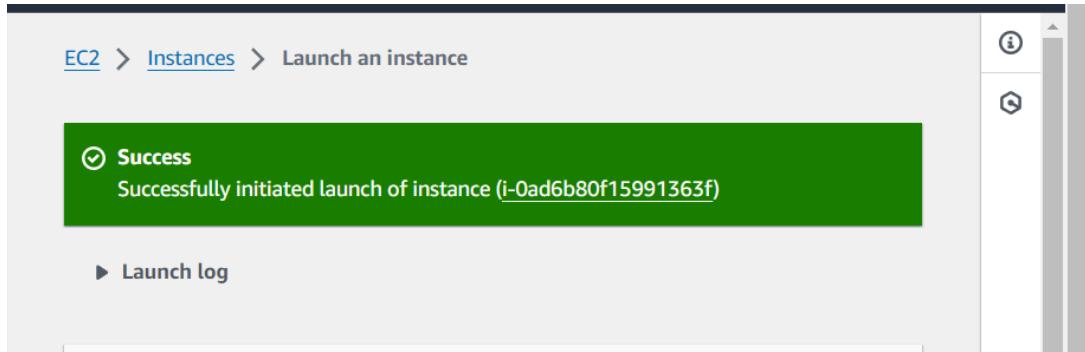
AWS

AWS

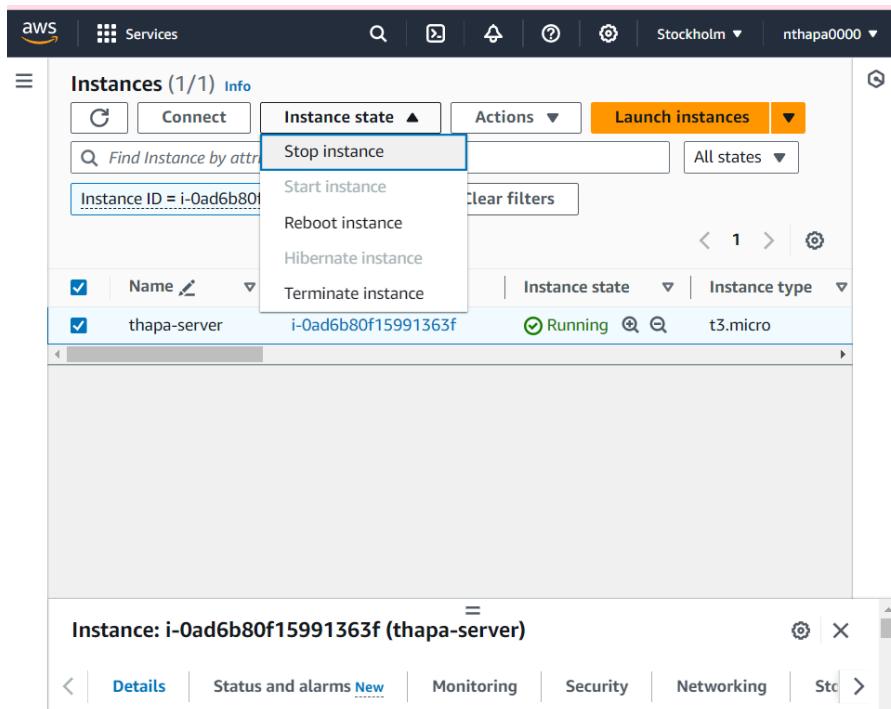
app.listen(80)
By default HTTP port is 80
By default HTTPS port is 443
HTTPS is more secure version of HTTP

node index.js
3000

http://1.2.3.4 ← → http://1.2.3.4:80



Remember to stop the instance after the use , or else money can be debited



Currently we have allowed only certain ports if we want to allow more ports we have to select the instance and click on security

And change the inbound rules

The screenshot shows the AWS CloudWatch Metrics console. At the top, there are tabs for 'Metrics' (selected), 'Logs', and 'CloudWatch Metrics Insights'. Below the tabs, there is a search bar and a filter section with dropdowns for 'Metric Name' (set to 'CPU'), 'Time range' (set to 'Last hour'), and 'Region' (set to 'All regions'). A large chart area displays a single data series for the 'CPU' metric, showing values fluctuating between 0% and 100% over the last hour. The chart has a light blue background with white grid lines.

The screenshot shows the AWS CloudWatch Metrics console. At the top, there are tabs for 'Metrics' (selected), 'Logs', and 'CloudWatch Metrics Insights'. Below the tabs, there is a search bar and a filter section with dropdowns for 'Metric Name' (set to 'CPU'), 'Time range' (set to 'Last hour'), and 'Region' (set to 'All regions'). A large chart area displays a single data series for the 'CPU' metric, showing values fluctuating between 0% and 100% over the last hour. The chart has a light blue background with white grid lines.

If we want to allow our application on different port we have to update the security group
Now i want to connect to the server

SSh into the Server

Connect to the server

Save the key pair file in a folder , then go to that folder

```
ssh -i thapa-password.pem ubuntu@13.60.17.53
```

The screenshot shows a VS Code interface with a terminal tab active. The terminal window displays an SSH session to an Ubuntu 22.04.4 LTS host. The session starts with a warning about host authenticity and a key fingerprint. It then prompts for confirmation to continue connecting. Once connected, it shows system information, including system load, memory usage, and network details. It also indicates that no updates are available immediately and provides instructions for enabling ESM Apps.

```
C:\Users\NTC\Desktop\Dev\Week11>ssh -i thapa-password.pem ubuntu@13.60.17.53
The authenticity of host '13.60.17.53 (13.60.17.53)' can't be established.
ED25519 key fingerprint is SHA256:73j0+hc/jVFx9wDs0Ldts5NExBDyiMKw1fPq+jH9JjE.

This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '13.60.17.53' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 6.5.0-1014-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

System information as of Thu Mar 28 07:36:34 UTC 2024

System load:  0.0          Processes:           99
Usage of /:   20.4% of 7.57GB  Users logged in:    0
Memory usage: 21%           IPv4 address for ens5: 172.31.40.25
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status
```

WARNING: UNPROTECTED PRIVATE KEY FILE

(our certificate file should be secure)

We get warning if a file can be accessed by several users ,ideally it should be restricted

Ssh means secure shell it is a protocol. Securely connecting to a shell in that machine.

chmod 700 thapa-password.pem

We wont be able to run this command in a cmd so we can run this command in WSL or git bash

```
Last login: Thu Mar 28 07:36:35 2024 from 103.47.14.53
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
```

```
ubuntu@ip-172-31-40-25:~$  
ubuntu@ip-172-31-40-25:~$  
ubuntu@ip-172-31-40-25:~$
```

FunctionMade* ↗ Run Testcases ⚡ 0 △ 0 ⌂ 0 🔍 Live Share ⓘ Go Live ✅ Prettier ⓘ

If we want to exit it we can **exit**

git clone <https://github.com/hkirat/sum-server>

```
ubuntu@ip-172-31-40-25:~$ git clone https://github.com/hkirat/sum-server
Cloning into 'sum-server'...
remote: Enumerating objects: 23, done.
remote: Counting objects: 100% (23/23), done.
remote: Compressing objects: 100% (15/15), done.
Receiving objects: 100% (23/23), 13.32 KiB | 909.00 KiB/s, done.
Resolving deltas: 100% (11/11), done.
remote: Total 23 (delta 11), reused 19 (delta 7), pack-reused 0
ubuntu@ip-172-31-40-25:~$ ls
sum-server
ubuntu@ip-172-31-40-25:~$
```

For some people instance can't access internet

💡 If your aws machine shows you the following error, your aws machine doesn't have access to the internet

Solution - <https://www.tecmint.com/resolve-temporary-failure-in-name-resolution/>

```
ubuntu@ip-172-31-11-253:~$ ping google.com
ping: google.com: Temporary failure in name resolution
ubuntu@ip-172-31-11-253:~$
```

ping google.com

How to deploy dockertized app in AWS?



Running for only our IP

Domain is constant, public IP keep on changing

```
ubuntu@ip-172-31-40-25:~$ ls
sum-server
ubuntu@ip-172-31-40-25:~$ cd sum-server
ubuntu@ip-172-31-40-25:~/sum-server$ ls
index.js  package-lock.json  package.json
ubuntu@ip-172-31-40-25:~/sum-server$
```

We have our node.js code in that aws machine located in my case Stockholm

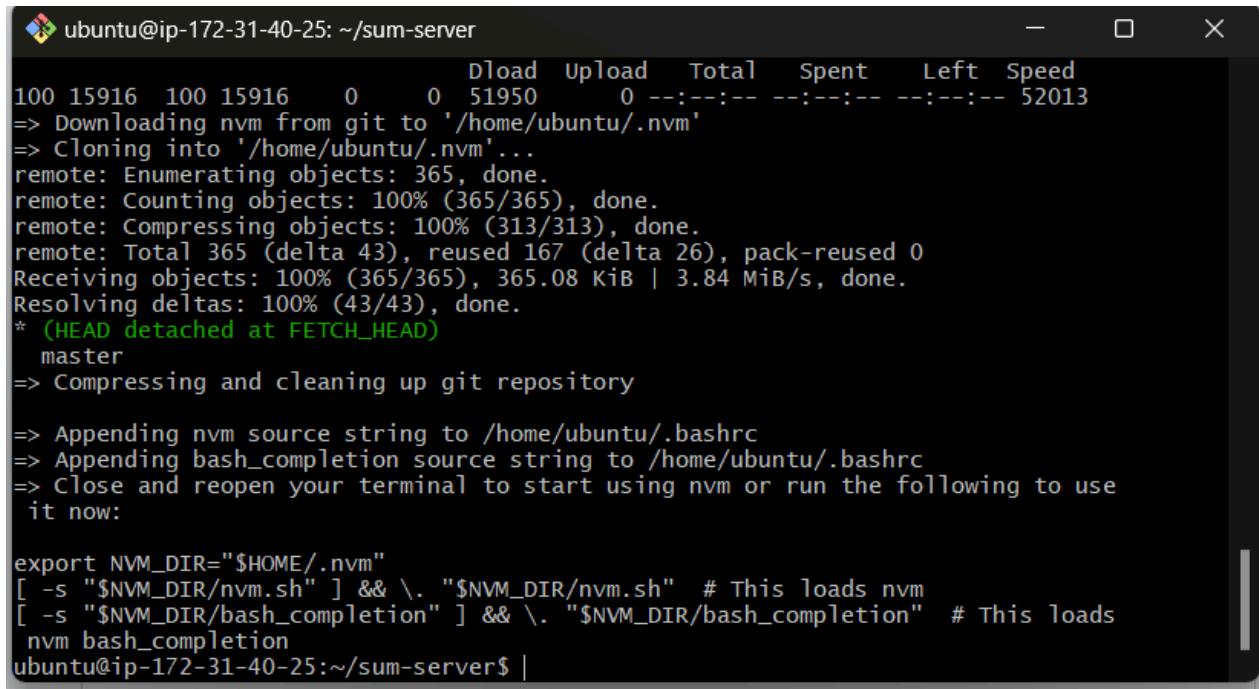
Install Node.js

(<https://www.digitalocean.com/community/tutorials/how-to-install-node-js-on-ubuntu-20-04>)

We will use NVM

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.3/install.sh | bash
```

Take the content and take it in bash



```
ubuntu@ip-172-31-40-25: ~/sum-server
Dload Upload Total Spent Left Speed
100 15916 100 15916 0 0 51950 0 --::-- --::-- --::-- 52013
=> Downloading nvm from git to '/home/ubuntu/.nvm'
=> Cloning into '/home/ubuntu/.nvm'...
remote: Enumerating objects: 365, done.
remote: Counting objects: 100% (365/365), done.
remote: Compressing objects: 100% (313/313), done.
remote: Total 365 (delta 43), reused 167 (delta 26), pack-reused 0
Receiving objects: 100% (365/365), 365.08 KiB | 3.84 MiB/s, done.
Resolving deltas: 100% (43/43), done.
* (HEAD detached at FETCH_HEAD)
 master
=> Compressing and cleaning up git repository
=> Appending nvm source string to /home/ubuntu/.bashrc
=> Appending bash_completion source string to /home/ubuntu/.bashrc
=> Close and reopen your terminal to start using nvm or run the following to use
it now:

export NVM_DIR="$HOME/.nvm"
[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh" # This loads nvm
[ -s "$NVM_DIR/bash_completion" ] && \. "$NVM_DIR/bash_completion" # This loads
nvm bash_completion
ubuntu@ip-172-31-40-25:~/sum-server$
```

Installing nvm

Now we have access to nvm

```
ubuntu@ip-172-31-40-25: ~/sum-server
nvm bash_completion
ubuntu@ip-172-31-40-25:~/sum-server$ export NVM_DIR="$HOME/.nvm"
[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh" # This loads nvm
[ -s "$NVM_DIR/bash_completion" ] && \. "$NVM_DIR/bash_completion"
ubuntu@ip-172-31-40-25:~/sum-server$ nvm

Node Version Manager (v0.39.3)

Note: <version> refers to any version-like string nvm understands. This includes:
  - full or partial version numbers, starting with an optional "v" (0.10, v0.1.2, v1)
  - default (built-in) aliases: node, stable, unstable, iojs, system
  - custom aliases you define with `nvm alias foo`'

Any options that produce colorized output should respect the `--no-colors` option.

Usage:
  nvm --help                                Show this message
  --no-colors                               Suppress colored output
  nvm --version                             Print out the installed version of nvm
  nvm install [<version>]                   Download and install a <version>. Uses .nv
mrc if available and version is omitted.
  The following optional arguments, if provided, must appear directly after `nvm instal
l`:
  -s                                         Skip binary download, install from source
only.
  -b                                         Skip source download, install from binary
only.
  --reinstall-packages-from=<version>
led in <node|iojs|node version number>
  --lts                                      When installing, reinstall packages instal
g-term support) versions
  --lts=<LTS name>
for a specific LTS line
  --skip-default-packages
file if it exists
  --latest-npm
e latest working npm on the given node version
  --no-progress                            Disable the progress bar on any downloads
```

Install all dependencies

Currently we are bringing node in our machine every time we are running something. Ideally we should containerize our app that way our machine don't need node.js only docker container need it.

nvm and nvim is different things.

```
ubuntu@ip-172-31-40-25:~/sum-server
Note:
  to remove, delete, or uninstall nvm - just remove the `'$NVM_DIR` folder (usually `~/.n
  vm`)
ubuntu@ip-172-31-40-25:~/sum-server$ nvm install 20
Downloading and installing node v20.12.0...
Downloading https://nodejs.org/dist/v20.12.0/node-v20.12.0-linux-x64.tar.xz...
#####
Computing checksum with sha256sum
Checksums matched!
Now using node v20.12.0 (npm v10.5.0)
Creating default alias: default -> 20 (-> v20.12.0)
ubuntu@ip-172-31-40-25:~/sum-server$ node
Welcome to Node.js v20.12.0.
Type ".help" for more information.
>
> (To exit, press Ctrl+C again or Ctrl+D or type .exit)
>
ubuntu@ip-172-31-40-25:~/sum-server$ npm
npm <command>

Usage:
  npm install      install all the dependencies in your project
  npm install <foo> add the <foo> dependency to your project
  npm test         run this project's tests
  npm run <foo>   run the script named <foo>
  npm <command> -h quick help on <command>
  npm -l          display usage info for all commands
  npm help <term> search for help on <term>
  npm help npm    more involved overview

All commands:
  access, adduser, audit, bugs, cache, ci, completion,
  config, dedupe, deprecate, diff, dist-tag, docs, doctor,
  edit, exec, explain, explore, find-dupes, fund, get, help,
  help-search, hook, init, install, install-ci-test,
```

Npm install

Node index.js

```
test, token, uninstall, unpublish, unstar, update, version,
view, whoami

Specify configs in the ini-formatted file:
  /home/ubuntu/.npmrc
or on the command line via: npm <command> --key=value

More configuration info: npm help config
Configuration fields: npm help 7 config

npm@10.5.0 /home/ubuntu/.nvm/versions/node/v20.12.0/lib/node_modules/npm
ubuntu@ip-172-31-40-25:~/sum-server$ npm install

added 64 packages, and audited 65 packages in 2s

11 packages are looking for funding
  run `npm fund` for details

1 moderate severity vulnerability

To address all issues, run:
  npm audit fix

Run `npm audit` for details.
ubuntu@ip-172-31-40-25:~/sum-server$ node index.js
```

Now our backend app is running in server

Something running in port 8080

cat index.js

To know the content of the file and check out in which port this is running

```
function getRandomNumber(max) {
    return Math.floor(Math.random() * max);
}

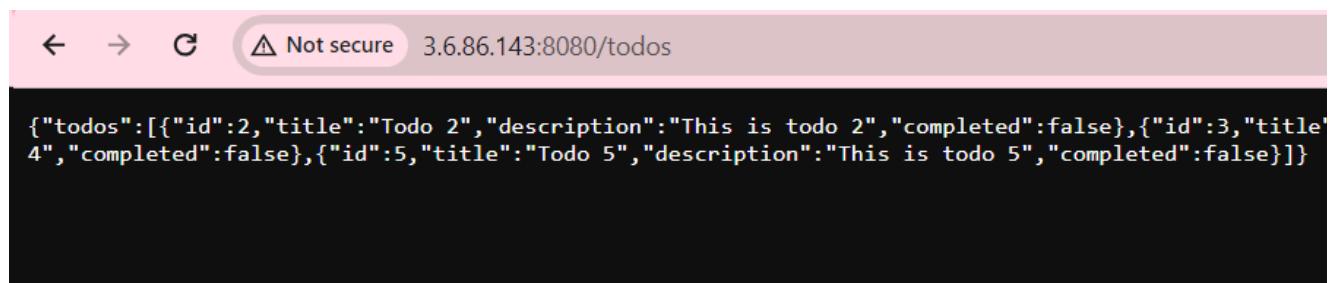
app.get("/notifications", (req, res) => {
    res.json({
        network: getRandomNumber(10),
        jobs: getRandomNumber(10),
        messaging: getRandomNumber(10),
        notifications: getRandomNumber(10)
    })
})

app.listen(8080)
ubuntu@ip-172-31-40-25:~/sum-server$
```

See that the index.js is running on port 8080

Now go to the AWS instance and allow port 8080

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-091ee559265de67e4	SSH	TCP	22	Custom	0.0.0.0/0
sgr-0d1c9f44e9db1053a	HTTP	TCP	80	Custom	0.0.0.0/0
sgr-0ce4f95a10bcb5e19	HTTPS	TCP	443	Custom	0.0.0.0/0
-	Custom TCP	TCP	8080	Anyw...	0.0.0.0/0
-	Custom TCP	TCP	8080	Anyw...	::/0



Reverse Proxy

We want ideally our frontend to run in port 80 so we don't have to pass any port ,
Hence we want to use reverse proxy

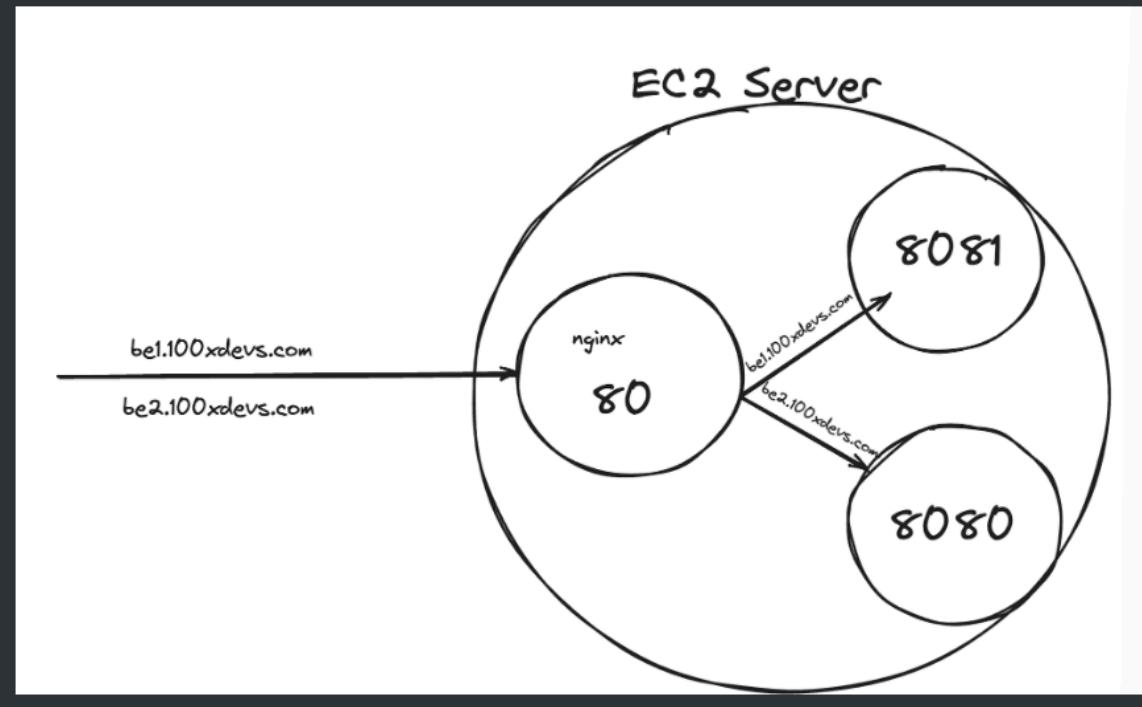
We are running something else in port 80 (called reverse proxy)(Not node.js process , its goal is to do opposite of proxy (in college network they block some websites, people add a proxy url (basically we are using third server to hit the url which is blocked))) and then according to url it will send that request to port 8081 or 8082

Reverse proxy is that inside the blocked website and do the reverse things
VPN are also proxy

<https://www.nginx.com/resources/glossary/nginx/>

NGINX is open source software for web serving, reverse proxying, caching, load balancing, media streaming, and more. It started out as a web server designed for maximum performance and stability. In addition to its HTTP server capabilities, NGINX can also function as a proxy server for email (IMAP, POP3, and SMTP) and a reverse proxy and load balancer for HTTP, TCP, and UDP servers.

What is a reverse proxy?



They will hit port 80

Default port get hit is port 80

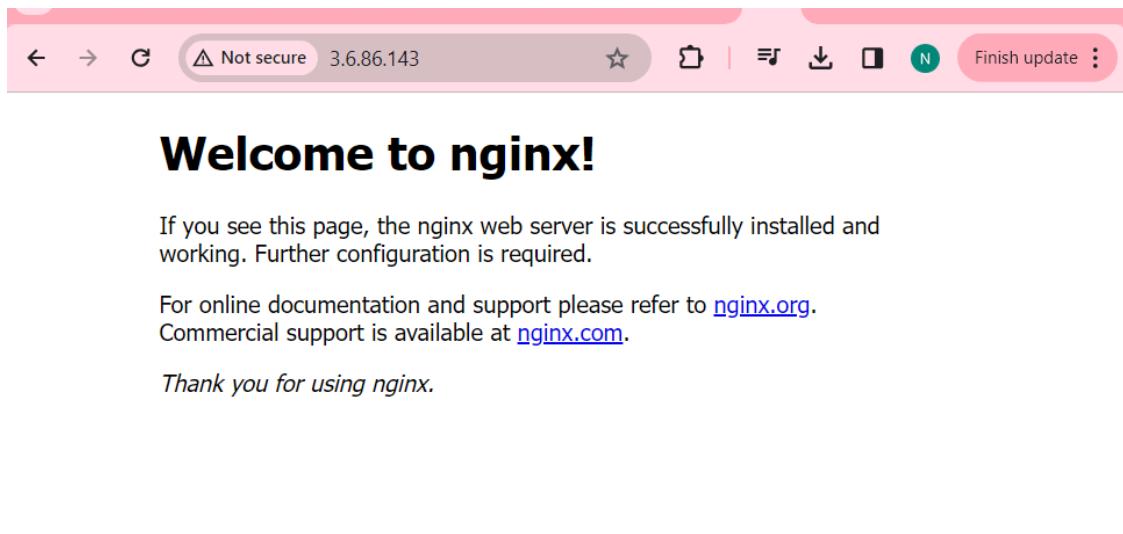
Default run on reverse proxy whose job is if client 1 is visiting here go there if client 2 is visiting go there

Install nginx in the virtual machine

sudo apt update

sudo apt install nginx

Now when we got to default port



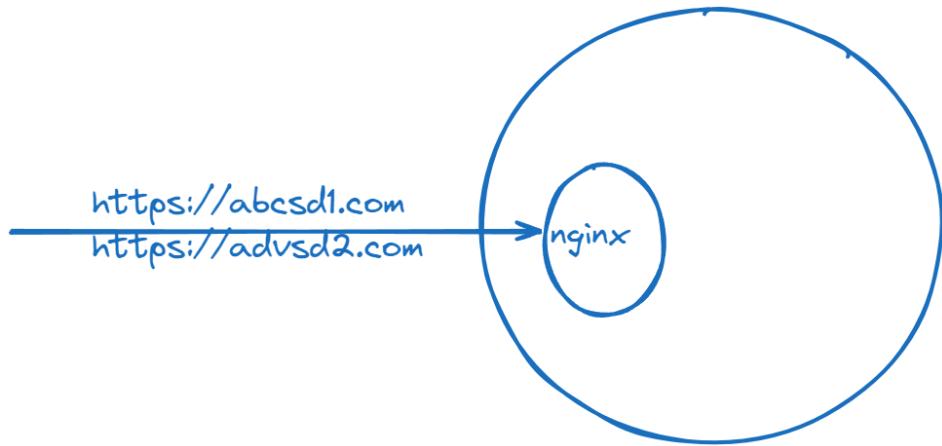
It is by default doing web serving or serving the HTML

Now we want it to do the reverse proxy.

Nginx is taking over the port 80, the request is being routed to nginx

Watch the Video from 1hr 38 min if you have sub-domain

To see how to point two different url to the same server.



How to give google.com which currently point to a public IP and make it point to a IP of our choice but only in our own device.