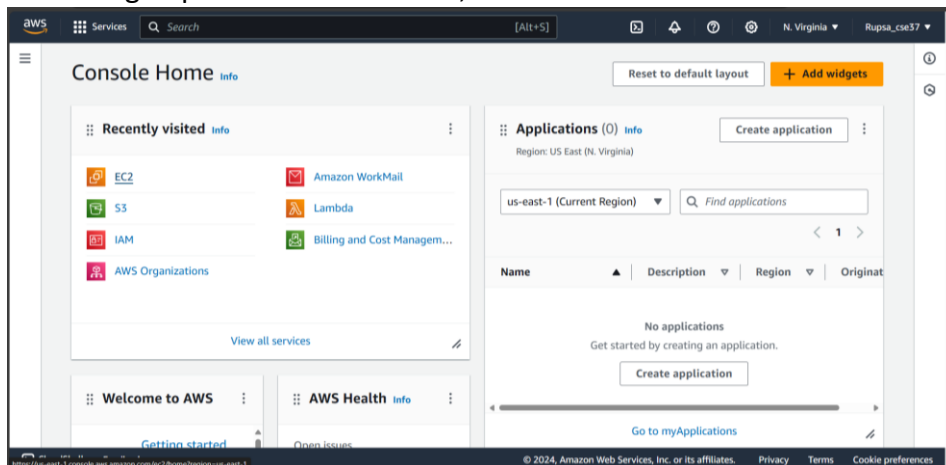


## PROBLEM STATEMENT :

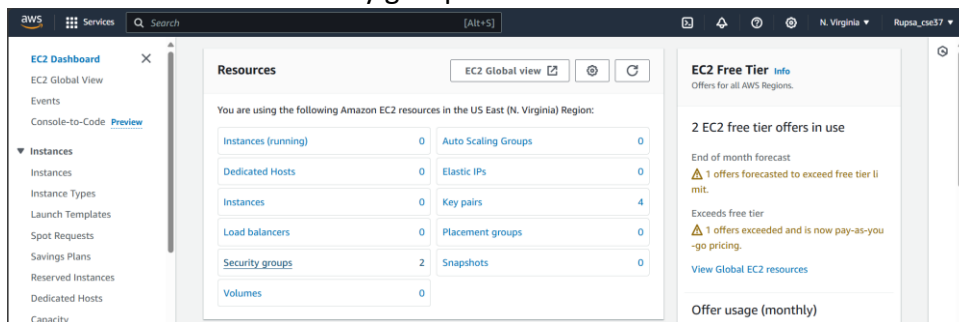
10) Deploy and run the project in AWS without using the port.

### Steps to deploy and run the project without using the port:

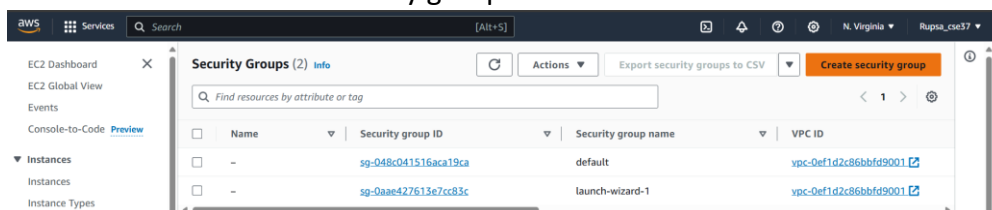
1. Sign up for an AWS account, search for 'EC2' then click on it.



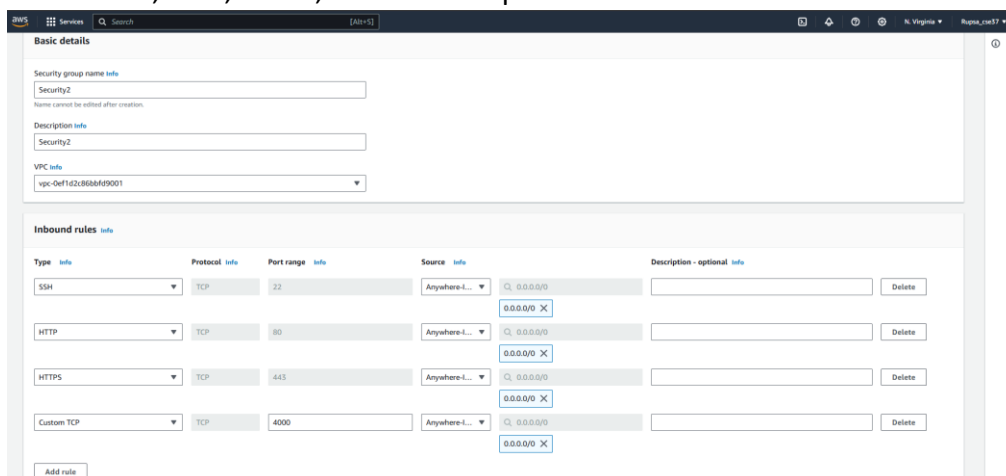
2. Now click on security group.



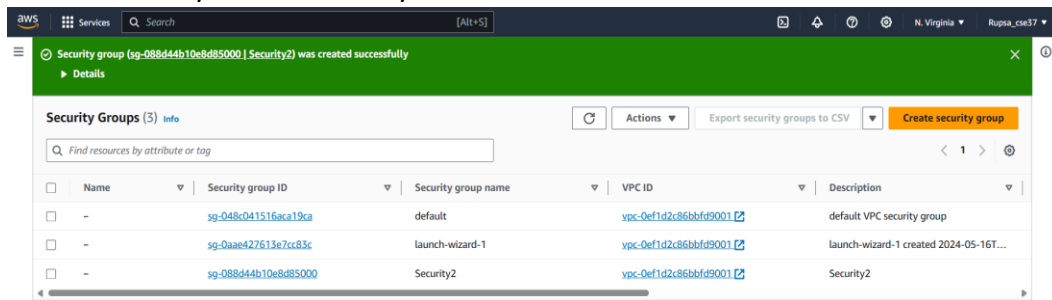
3. Click on "Create security group".



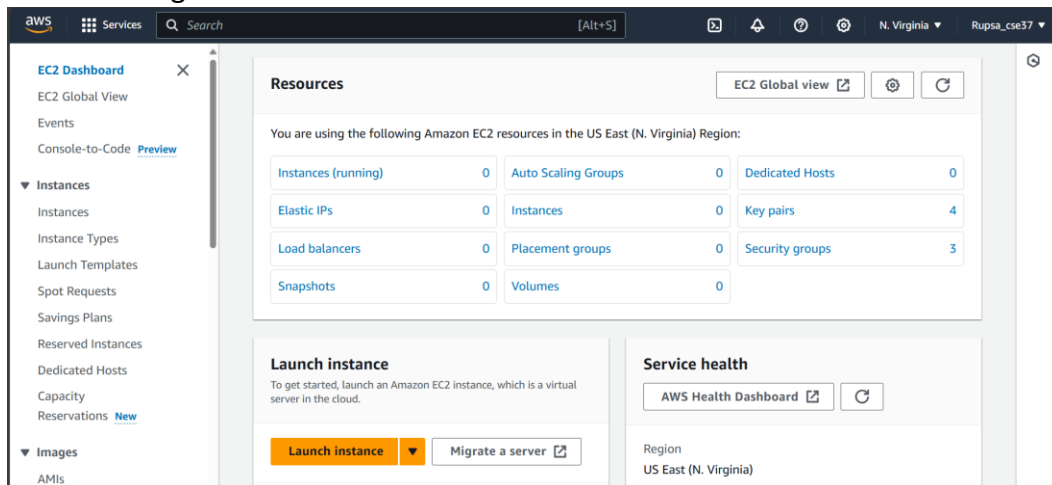
4. Give the name then in "Inbound rules" click on "Add rule" & add SSH,HTTP,HTTPS,CUSTOM TCP protocol and then click on "Create security group".



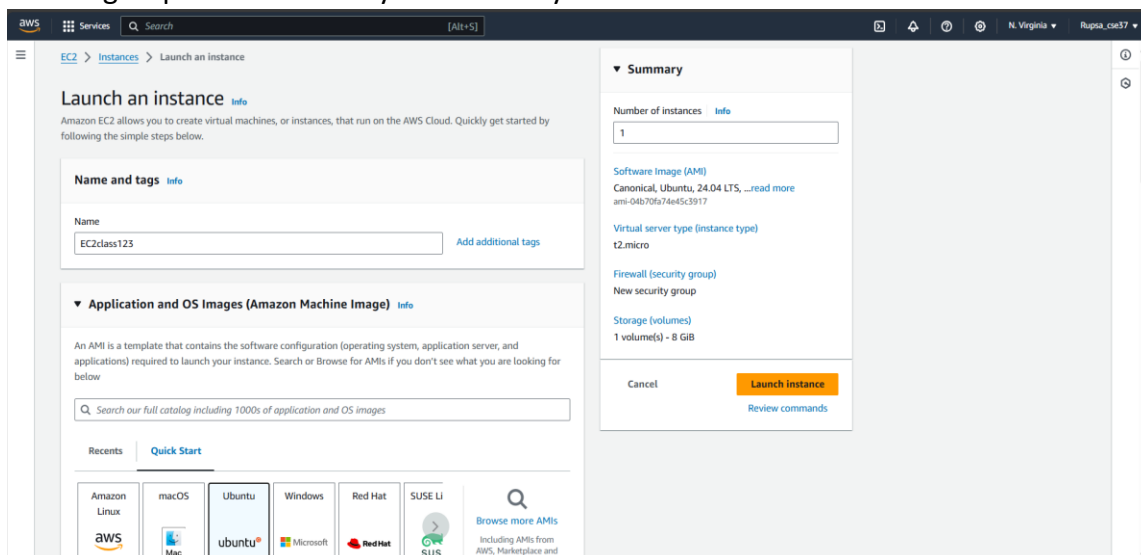
5. Security2 is successfully created.



6. Now go back to EC2 dashboard & click on “Launch Instance”.



7. Now write the name, choose the platform, keypair and click on “Select existing security group” that is Security2. Then finally click on “Launch Instance”.



**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  
key1 [Create new key pair](#)

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

Network Info  
vpc-0ef1d2c8b6bf9001

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

Common security groups Info  
Select security groups  
Security2 sg-088d44b10e8d85000 [Compare security group rules](#)

**Summary**

Number of instances Info  
1

Software Image (AMI)  
Canonical, Ubuntu, 24.04 LTS, ...read more  
ami-04b70fa74e45c3917

Virtual server type (instance type)  
t2.micro

Firewall (security group)  
Security2

Storage (volumes)  
1 volume(s) - 8 GiB

[Cancel](#) [Launch instance](#) [Review commands](#)

8. The instance is successfully created and select it then copy the Public IPv4 address & paste it in Bitwise.

**Instances (1/1)** Info [Refresh](#) [Connect](#) [Instance state](#) [Actions](#) [Launch instances](#)

☒ Find Instance by attribute or tag (case-sensitive) [All states](#) [1](#)

<input checked="" type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input checked="" type="checkbox"/>	EC2class123	i-08cf91e6d225b9b04	Running	t2.micro	-	<a href="#">View alarms</a>	us-east-1b

**i-08cf91e6d225b9b04 (EC2class123)**

[Details](#) [Status and alarms](#) [Monitoring](#) [Security](#) [Networking](#) [Storage](#) [Tags](#)

**Instance summary** Info

Instance ID  
i-08cf91e6d225b9b04 (EC2class123)

IPv6 address  
-

Public IPv4 address copied  
54.234.56.173 | [open address](#)

Instance state  
Running

Private IPv4 addresses  
172.31.20.241

Public IPv4 DNS  
ec2-54-234-56-173.compute-1.amazonaws.com | [open address](#)

9. Now In Bitwise paste it and import the key from Client key manager then click on “Log in”.

Bitwise SSH Client 9.33 [Window behavior](#)

**Default profile**

[Load profile](#) [Save profile as](#) [New profile](#) [Reset profile](#)

[Login](#) [Options](#) [Terminal](#) [RDP](#) [SFTP](#) [Services](#) [C2S](#) [S2C](#) [SSH](#) [Notes](#) [About\\*](#)

**Server**

Host 54.234.56.173

Port  ☐ Enable obfuscation

Obfuscation keyword

**Authentication**

Username ubuntu

Initial method publickey

Client key Global 1

Passphrase

Elevation Default

**Kerberos**

SPN

☐ GSS/Kerberos key exchange

☐ Request delegation

☒ gssapi-keyex authentication

[Proxy settings](#) [Host key manager](#) [Client key manager](#) [Help](#)

18:35:03.100 Current date: 2024-05-20

18:35:03.100 Bitwise SSH Client 9.33, a fully featured SSH client for Windows.

18:35:03.100 Copyright (C) 2009-2023 by Bitwise Limited.

18:35:03.100 Visit www.bitwise.com for latest information about our SSH software.

18:35:03.100 Run 'bVssh -help' to learn about supported command-line parameters.

18:35:03.100 Cryptographic provider: Windows CNG (x86) with additions

18:35:03.387 Optional update available.

18:35:03.435 Loading default profile.

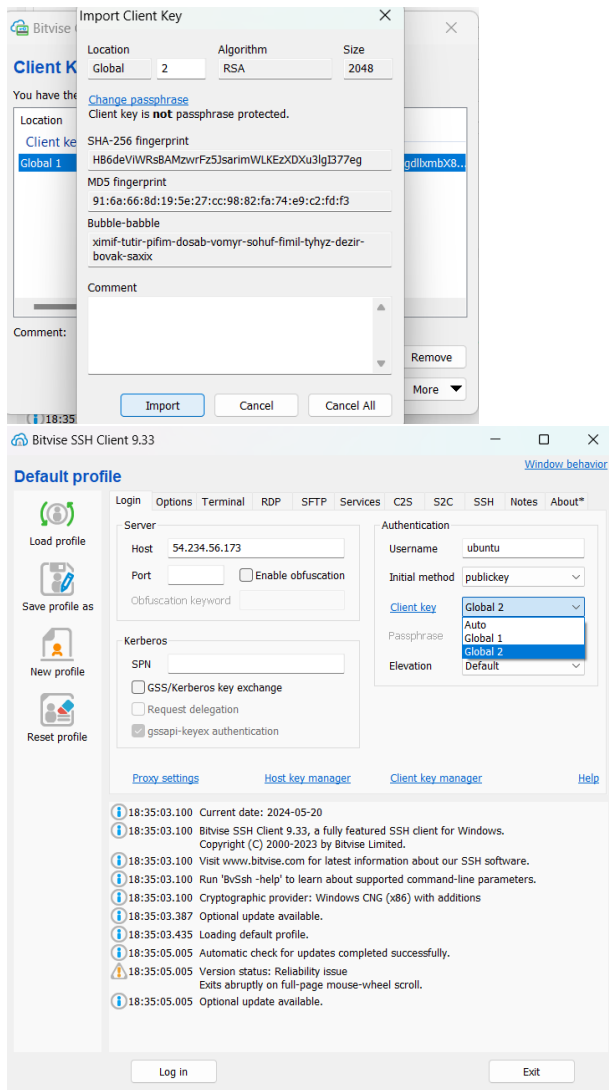
18:35:05.005 Automatic check for updates completed successfully.

18:35:05.005 Version status: Reliability issue

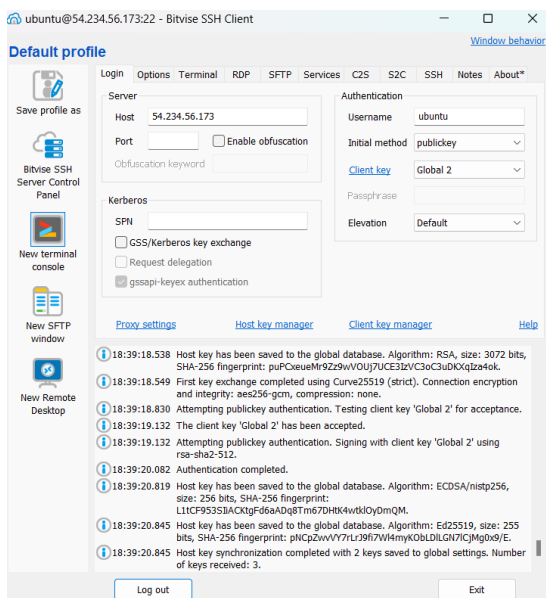
18:35:05.005 Exits abruptly on full-page mouse-wheel scroll.

18:35:05.005 Optional update available.

[Log in](#) [Exit](#)



10. After successful log in click on “New terminal console” and write the following commands.



```

ubuntu@ip-172-31-20-241:~$ sudo apt-get update
Get:34 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 c-n-f Metadata [116 B]
Fetched 28.3 MB in 5s (5305 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-20-241:~$ sudo apt-get upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done

User sessions running outdated binaries:
  ubuntu @ session #3: sshd[1012,1123]
  ubuntu @ user manager service: systemd[1017]

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-20-241:~$ sudo apt-get install nginx
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  nginx-common
Suggested packages:
  fcgiwrap nginx-doc ssl-cert
The following NEW packages will be installed:
  nginx nginx-common
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 552 kB of archives.
After this operation, 1596 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y

User sessions running outdated binaries:
  ubuntu @ session #3: sshd[1012,1123]
  ubuntu @ user manager service: systemd[1017]

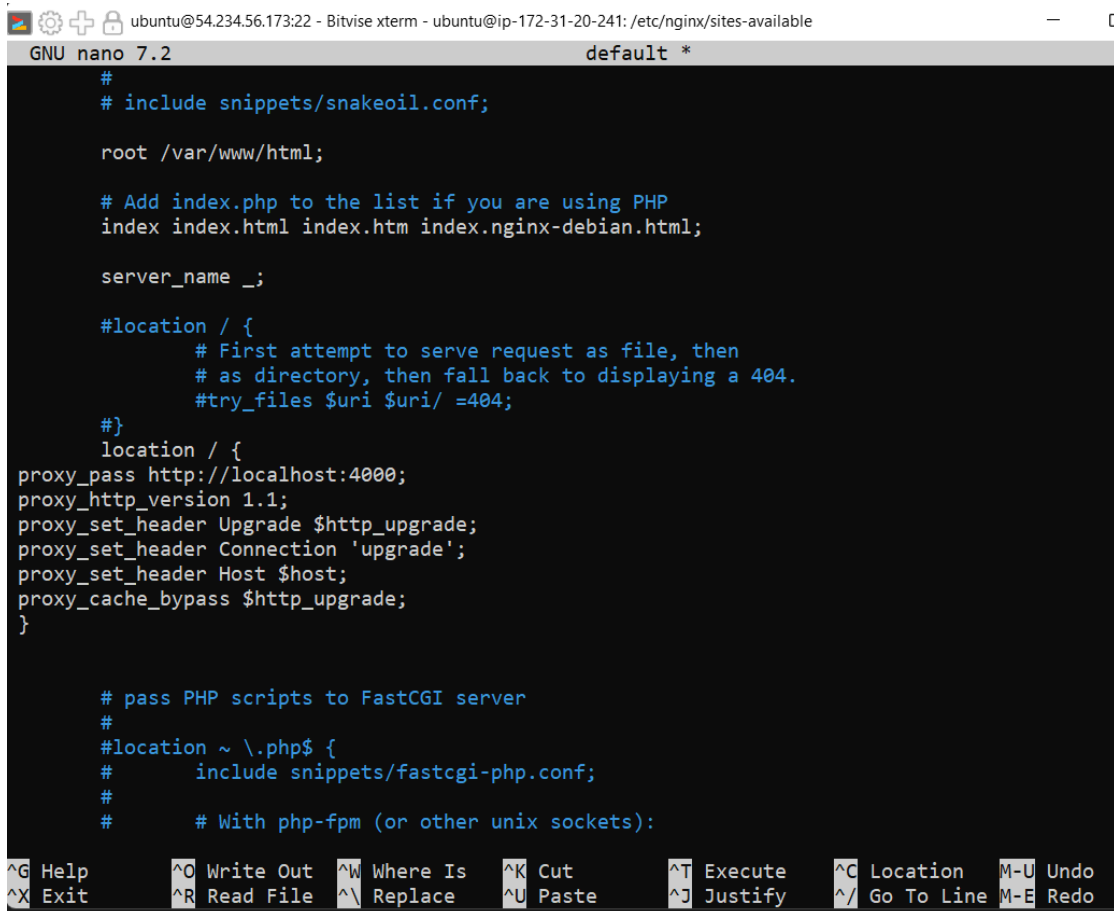
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-20-241:~$ nginx -v
nginx version: nginx/1.24.0 (Ubuntu)
ubuntu@ip-172-31-20-241:~$ curl -SL https://deb.nodesource.com/setup_18.x|sudo -E bash -
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  3154  100  3154    0     0  48058      0 --:--:-- --:--:-- --:--:-- 48523
2024-05-20 13:22:42 - Installing pre-requisites
Reading package lists... Done
2024-05-20 13:22:49 - Repository configured successfully. To install Node.js, run: apt-get install nodejs -y
ubuntu@ip-172-31-20-241:~$ sudo apt-get install nodejs
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-20-241:~$ nodejs -v
v18.20.2
ubuntu@ip-172-31-20-241:~$ cd /
ubuntu@ip-172-31-20-241:/$ pwd
/
ubuntu@ip-172-31-20-241:/$ sudo git clone https://github.com/Rupsa1037/Rup_AWS1.git
Cloning into 'Rup_AWS1'...
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 10 (delta 2), reused 5 (delta 0), pack-reused 0
Receiving objects: 100% (10/10), 49.08 KiB | 7.01 MiB/s, done.
Resolving deltas: 100% (2/2), done.
ubuntu@ip-172-31-20-241:/$ ls
Rup_AWS1  bin.usr-is-merged  dev  home  lib.usr-is-merged  lost+found  mnt  proc  run
bin       boot              etc  lib  lib64              media       opt  root /sbin
ubuntu@ip-172-31-20-241:/$ cd Rup_AWS1/
ubuntu@ip-172-31-20-241:/Rup_AWS1$ ls
'New Text Document.txt'  index.js  package-lock.json  package.json
ubuntu@ip-172-31-20-241:/Rup_AWS1$ sudo npm i
ubuntu@ip-172-31-20-241:/Rup_AWS1$ npm -v
10.5.0
ubuntu@ip-172-31-20-241:/Rup_AWS1$ node index.js
Started server

```

11. Then stop the server and open another “New terminal console’ & write the following commands.

```
Last login: Mon May 20 13:11:25 2024 from 202.78.234.236
ubuntu@ip-172-31-20-241:~$ cd /
ubuntu@ip-172-31-20-241:/$ pwd
/
ubuntu@ip-172-31-20-241:/$ cd etc/nginx/sites-available/
ubuntu@ip-172-31-20-241:/etc/nginx/sites-available$ sudo nano default
```



```
GNU nano 7.2 default *
#
# include snippets/snakeoil.conf;

root /var/www/html;

# Add index.php to the list if you are using PHP
index index.html index.htm index.nginx-debian.html;

server_name _;

#location / {
#    First attempt to serve request as file, then
#    as directory, then fall back to displaying a 404.
#    try_files $uri $uri/ =404;
#}
location / {
    proxy_pass http://localhost:4000;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection 'upgrade';
    proxy_set_header Host $host;
    proxy_cache_bypass $http_upgrade;
}

# pass PHP scripts to FastCGI server
#
#location ~ \.php$ {
#    include snippets/fastcgi-php.conf;
#
#    # With php-fpm (or other unix sockets):
```

```
ubuntu@ip-172-31-20-241:/etc/nginx/sites-available$ sudo systemctl restart nginx
ubuntu@ip-172-31-20-241:/etc/nginx/sites-available$
```

Then on the another terminal start the server by “node index.js”.

```
ubuntu@ip-172-31-20-241:/Rup_AWS1$ node index.js
Started server
^C
ubuntu@ip-172-31-20-241:/Rup_AWS1$ node index.js
Started server
```

12. Paste the IPv4 in the browser to host the file.

