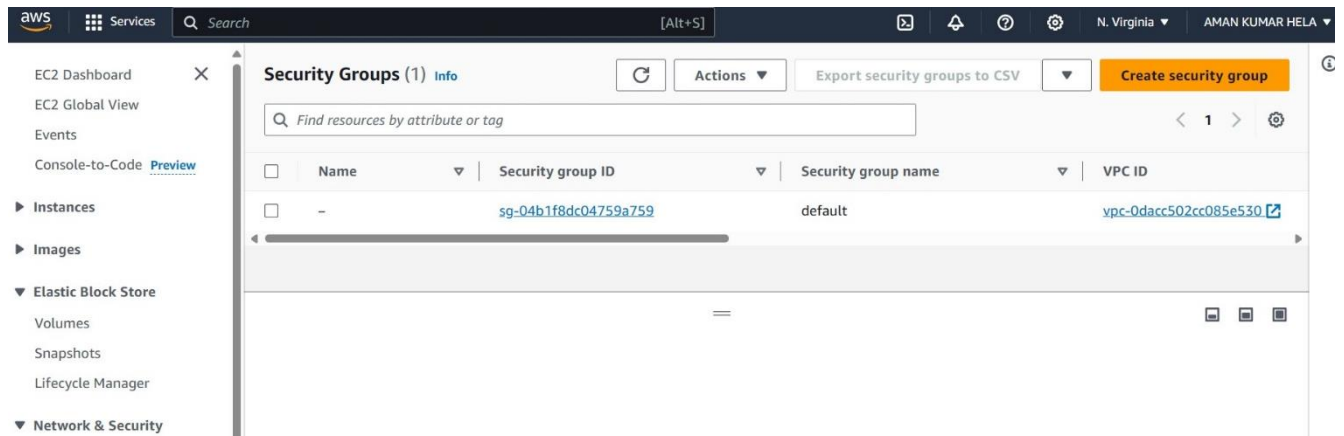


## Assignment-12:

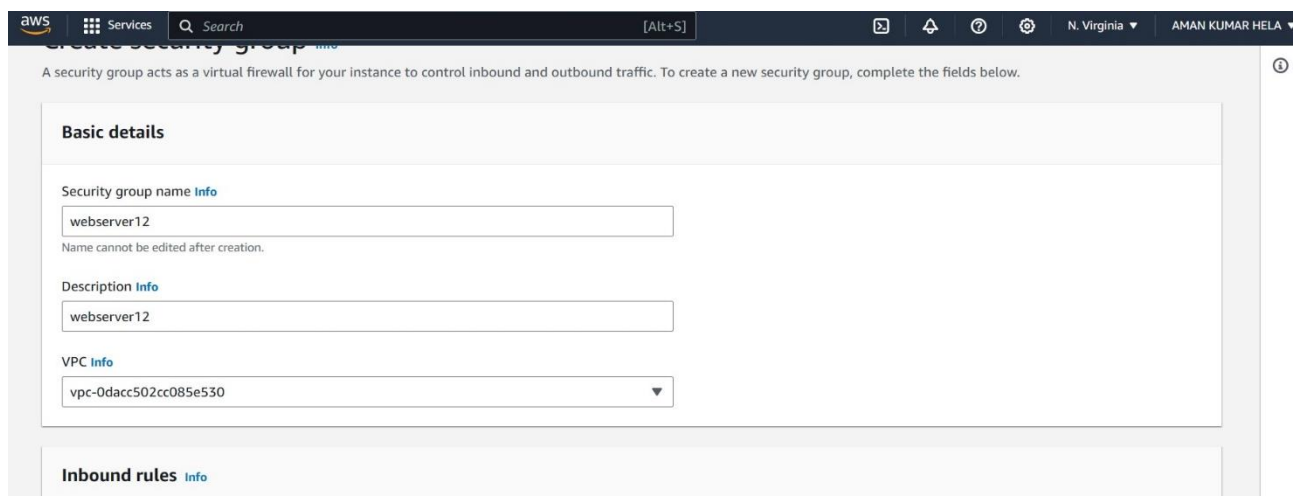
Deploy and run the project in AWS without using port.

Steps:

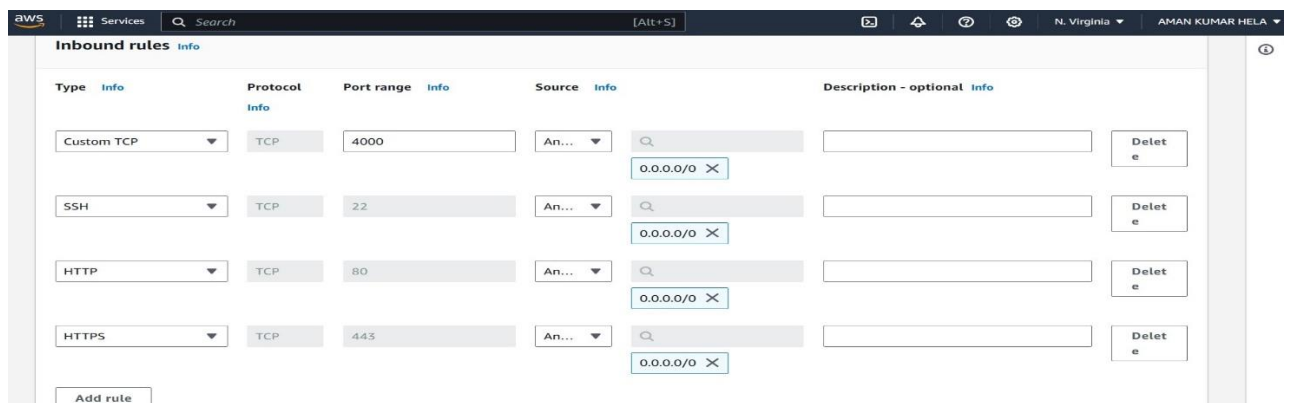
1. At first go to EC2 and then click on Create security group.



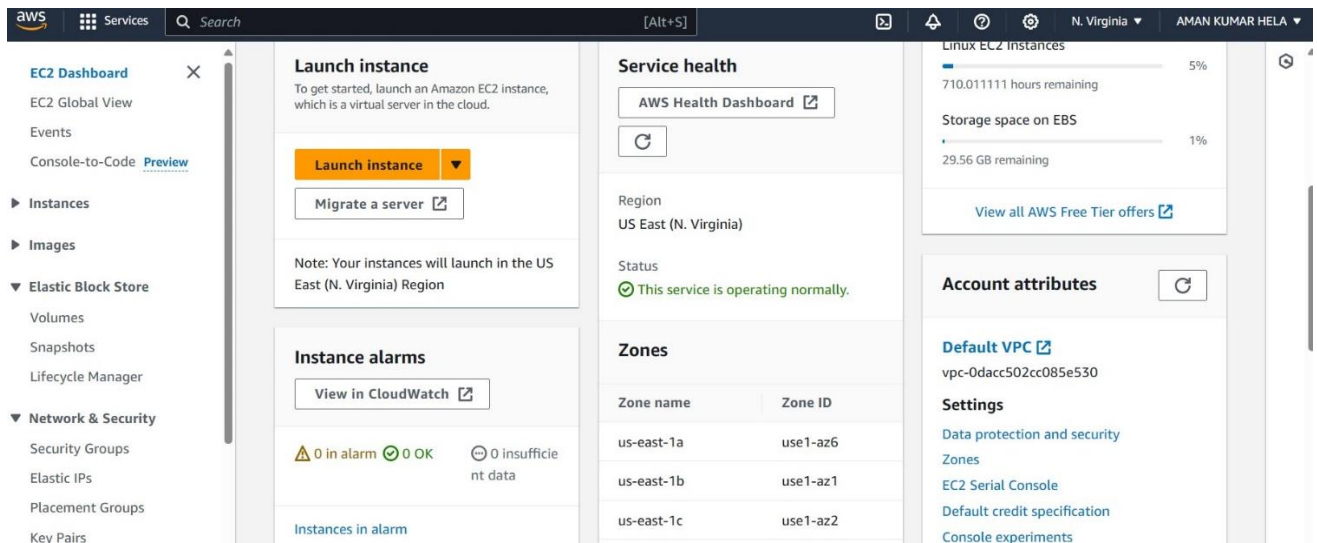
2. Now give its proper name and description.



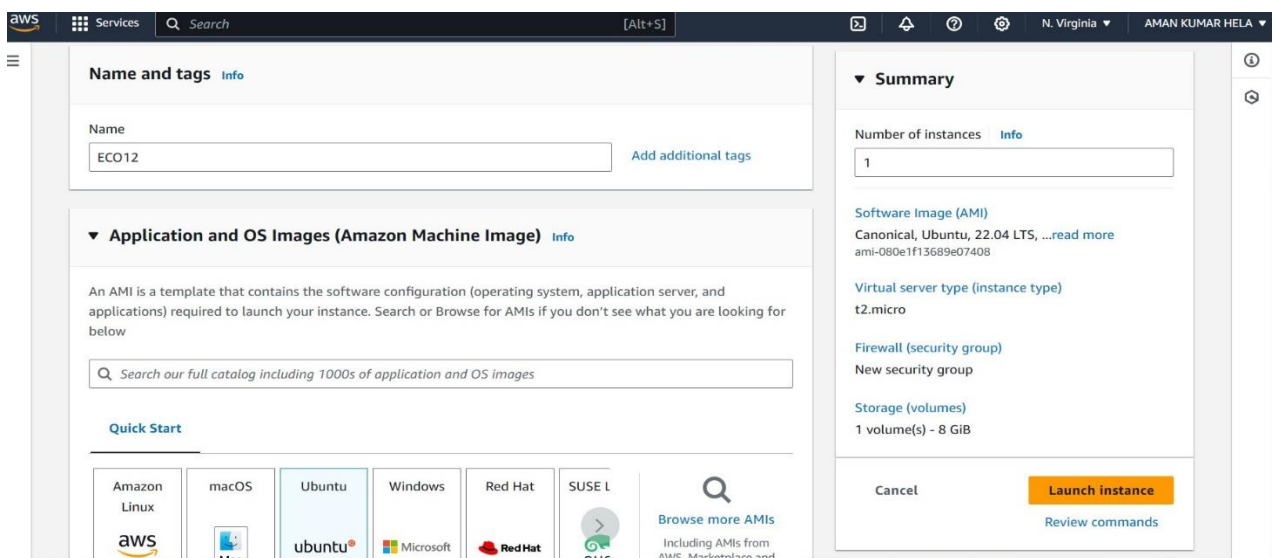
3. Now in inbound click on Add rule and in this way add 4 security rules.



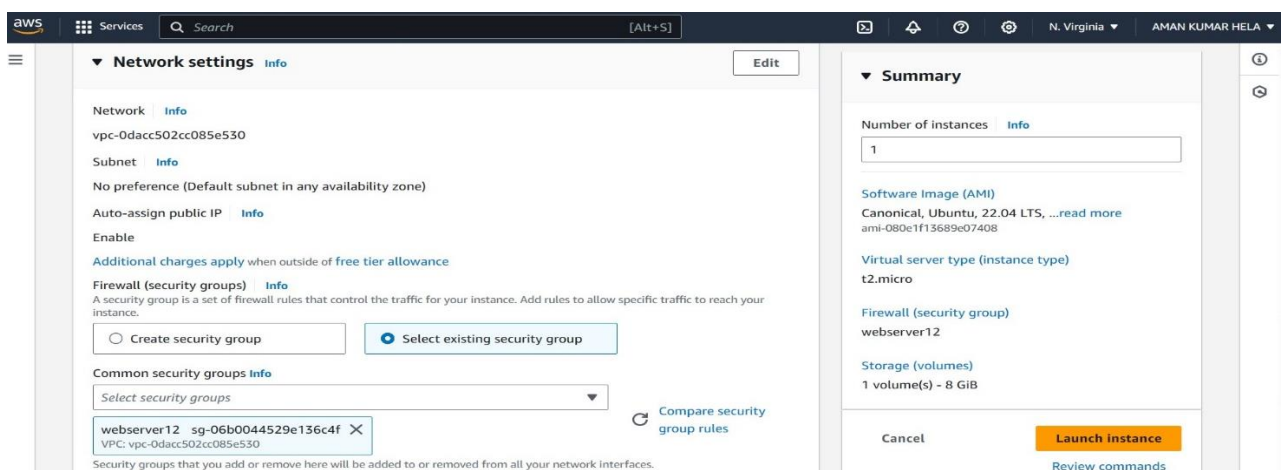
4. Now click on Create Security group and then go to Launch Instance.



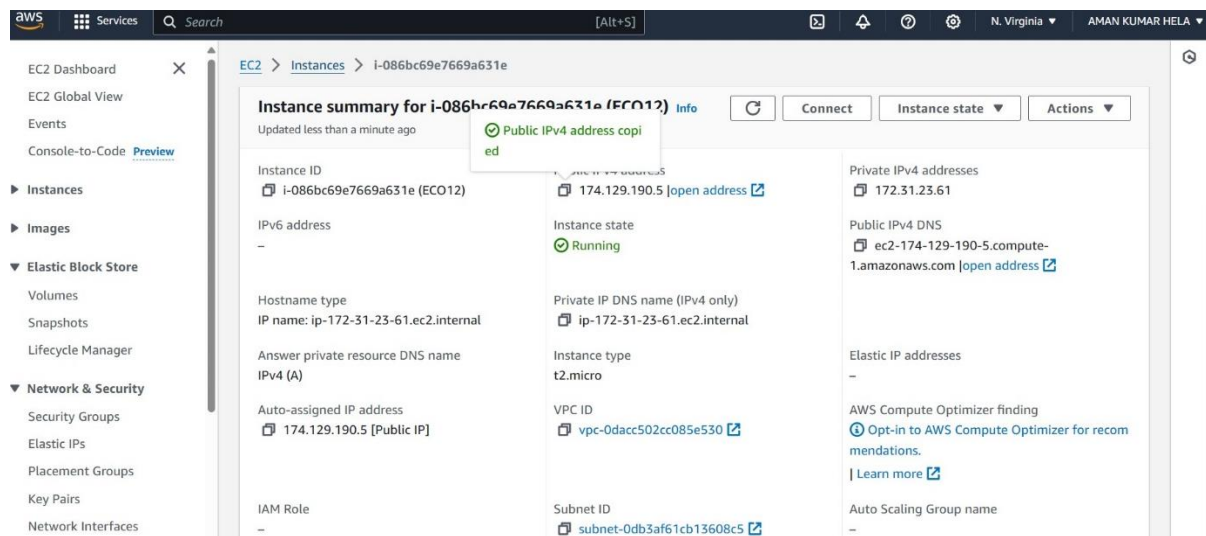
5. After it give name and then click on Ubuntu.



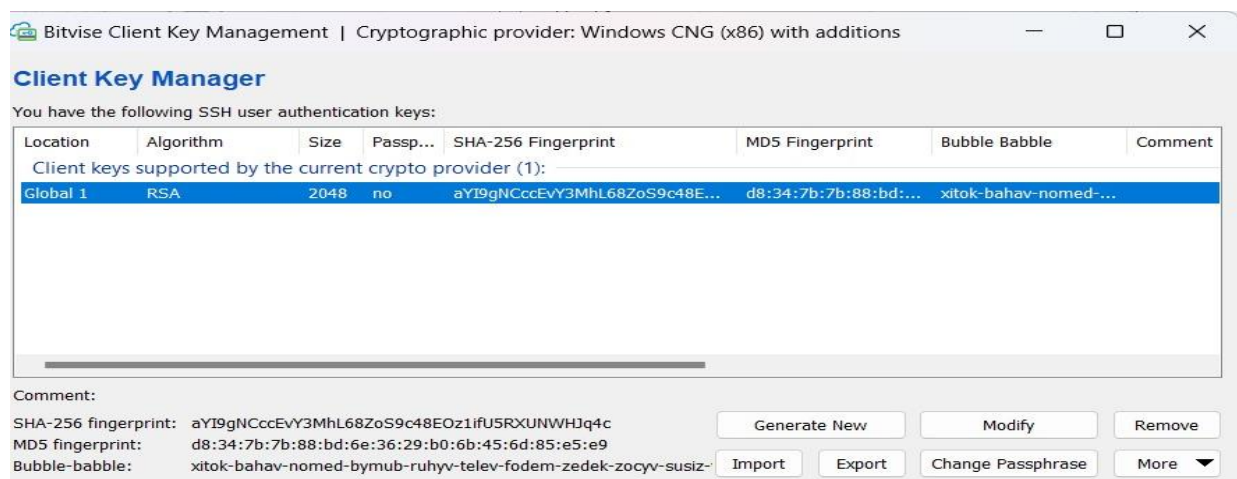
6. Now select key pair and also click on Select existing security group and click on Launch Instance.



7. Click on instance and copy public IPv4 address.



8. Paste it in host of BitVise SSH client and import key pair in Client key pair Manager.



9. Now open terminal after login and then write all commands :

- PWD

```
ubuntu@ip-172-31-23-61:~$ pwd
/home/ubuntu
ubuntu@ip-172-31-23-61:~$
```

- sudo apt-get update

```
ubuntu@ip-172-31-23-61:~$ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Fetched 110 kB in 1s (155 kB/s)
Reading package lists... Done
```



- `sudo apt-get upgrade`

```

Reading package lists... Done
ubuntu@ip-172-31-23-61:~$ sudo apt-get upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages have been kept back:
  linux-aws linux-headers-aws linux-image-aws python3-update-manager ubuntu-advantage-tools
  ubuntu-pro-client-l10n update-manager-core
The following packages will be upgraded:
  apt apt-utils bash bsdxtrautils bsdxutils coreutils curl dpkg eject ethtool fdisk klibc-utils
  libapt-pkg6.0 libblkid1 libc-bin libc6 libcurl3-gnutls libcurl4 libexpat1 libfdisk1 libgnutls30
  libgpgme11 libklibc libldap-2.5-0 libldap-common libmount1 libnspr4 libnss3 libsmartcols1
  libuuid1 locales mount openssl-client openssl-server openssl-sftp-server python3-cryptography

```

- `sudo apt-get install nginx`

```

ubuntu@ip-172-31-23-61:~$ sudo apt-get install nginx
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  fontconfig-config fonts-dejavu-core libdeflate0 libfontconfig1 libgd3 libjpeg-turbo8
  libjpeg8 libnginx-mod-http-geoip2 libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter
  libnginx-mod-mail libnginx-mod-stream libnginx-mod-stream-geoip2 libtiff5 libwebp7 libxpm4
  nginx-common nginx-core
Suggested packages:
  libgd-tools fcgiwrap nginx-doc ssl-cert
The following NEW packages will be installed:
  fontconfig-config fonts-dejavu-core libdeflate0 libfontconfig1 libgd3 libjpeg-turbo8

```

- `curl -SL https://deb.nodesource.com/setup_16.x|sudo -E bash -`

```

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-23-61:~$ curl -SL https://deb.nodesource.com/setup_16.x | sudo -E bash -
bash: -: No such file or directory
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
 72 5617    72 4098    0     0  47305      0 --:--:-- --:--:-- --:--:-- 47651

```

- `sudo apt install nodejs`

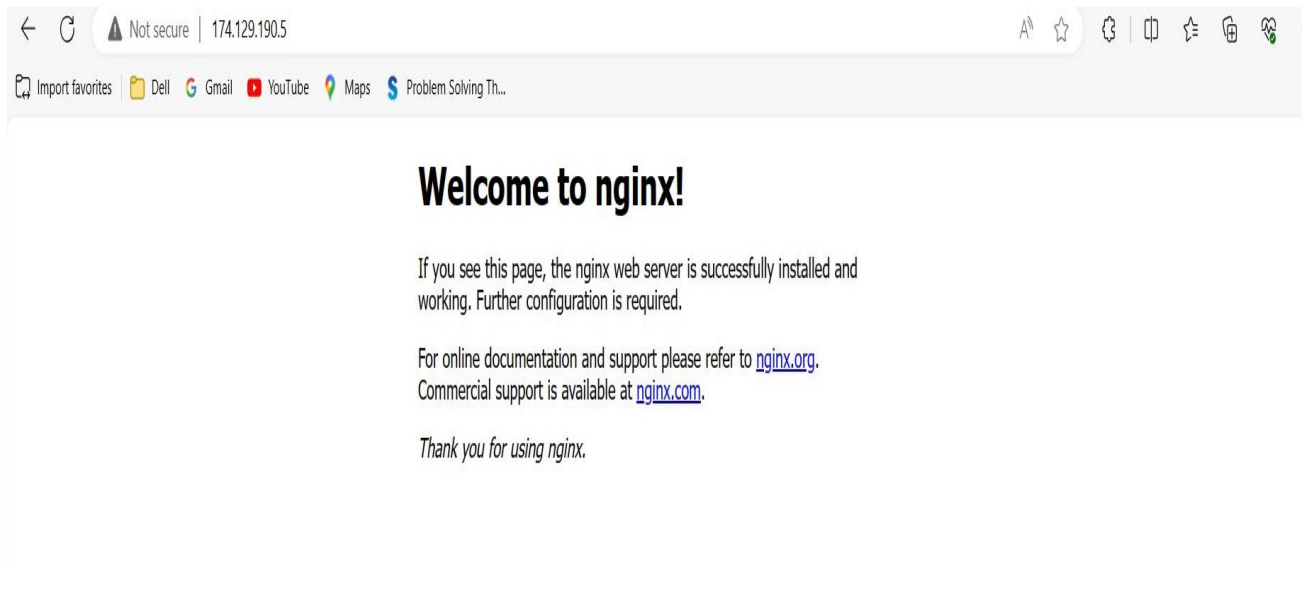
```
ubuntu@ip-172-31-23-61:~$ sudo apt install nodejs
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  javascript-common libc-ares2 libjs-highlight.js libnode72 nodejs-doc
Suggested packages:
  npm
The following NEW packages will be installed:
  javascript-common libc-ares2 libjs-highlight.js libnode72 nodejs nodejs-doc
0 upgraded, 6 newly installed, 0 to remove and 7 not upgraded.
Need to get 13.7 MB of archives.
After this operation, 54.0 MB of additional disk space will be used.
```

- `git clone`

<https://github.com/Amankr828/myrepo.git>

- `cd myrepo`
- `npm install`
- `node index.js`

10. Now server has started. If we paste it in url section then we can see nginx has started. To stop server click (ctrl+c).



```
Run `npm audit` for details.
ubuntu@ip-172-31-23-61:~/myrepo$ node index.js
Started server
```

11. Now write these all commands:

- `cd /`
- `pwd`
- `cd etc/nginx/sites-available/`

```
ubuntu@ip-172-31-23-61:~/myrepo$ cd /
ubuntu@ip-172-31-23-61:/$ pwd /
/
ubuntu@ip-172-31-23-61:/$ cd etc/nginx/sites-available/
ubuntu@ip-172-31-23-61:/etc/nginx/sites-available$
```

- `sudo nano default`

12. A new window will be opened. There at first go to location area and comment all codes and then write:

- `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;`  
  
`}`

```
# SSL configuration
#
# listen 443 ssl default_server;
# listen [::]:443 ssl default_server;
#
# Note: You should disable gzip for SSL traffic.
# See: https://bugs.debian.org/773332
#
# Read up on ssl_ciphers to ensure a secure configuration.
# See: https://bugs.debian.org/765782
#
# Self signed certs generated by the ssl-cert package
# Don't use them in a production server!
#
# 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/ {
    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;
}

# First attempt to serve request as file, then
# as directory, then fall back to displaying a 404.
try_files $uri / =404;
}

# pass PHP scripts to FastCGI server
#
```

Help Write Out Where Is Cut Execute Location M-U Undo M-A Set Mark M-T To Bracket M-Q Previous  
Exit Read File Replace Paste Justify Go To Line M-E Redo M-E Copy M-Q Where Was M-K Next

13. After it click ctrl+x, then y then click enter.

```
# listen 443 ssl default_server;
# listen [::]:443 ssl default_server;
#
# Note: You should disable gzip for SSL traffic.
# See: https://bugs.debian.org/773332
#
# Read up on ssl_ciphers to ensure a secure configuration.
# See: https://bugs.debian.org/765782
#
# Self signed certs generated by the ssl-cert package
# Don't use them in a production server!
#
# 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 / {
    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;
}

# First attempt to serve request as file, then
# as directory, then fall back to displaying a 404.
try_files $uri $uri/ =404;
}

# pass PHP scripts to FastCGI server
#
```

14. Now open new server terminal and write cd repo2.

15. Write sudo systemctl restart nginx

16. Now copy that public IPv4 address again and paste it in url and there you can see that without giving port(:4000) with url we have hosted the website.

