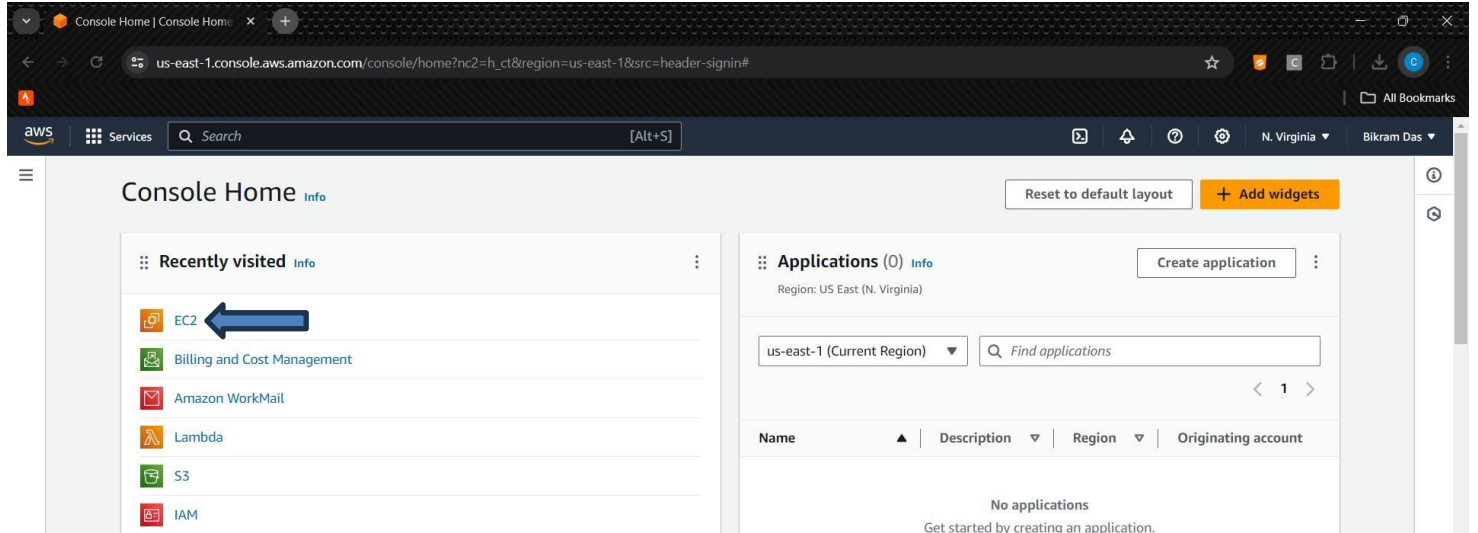


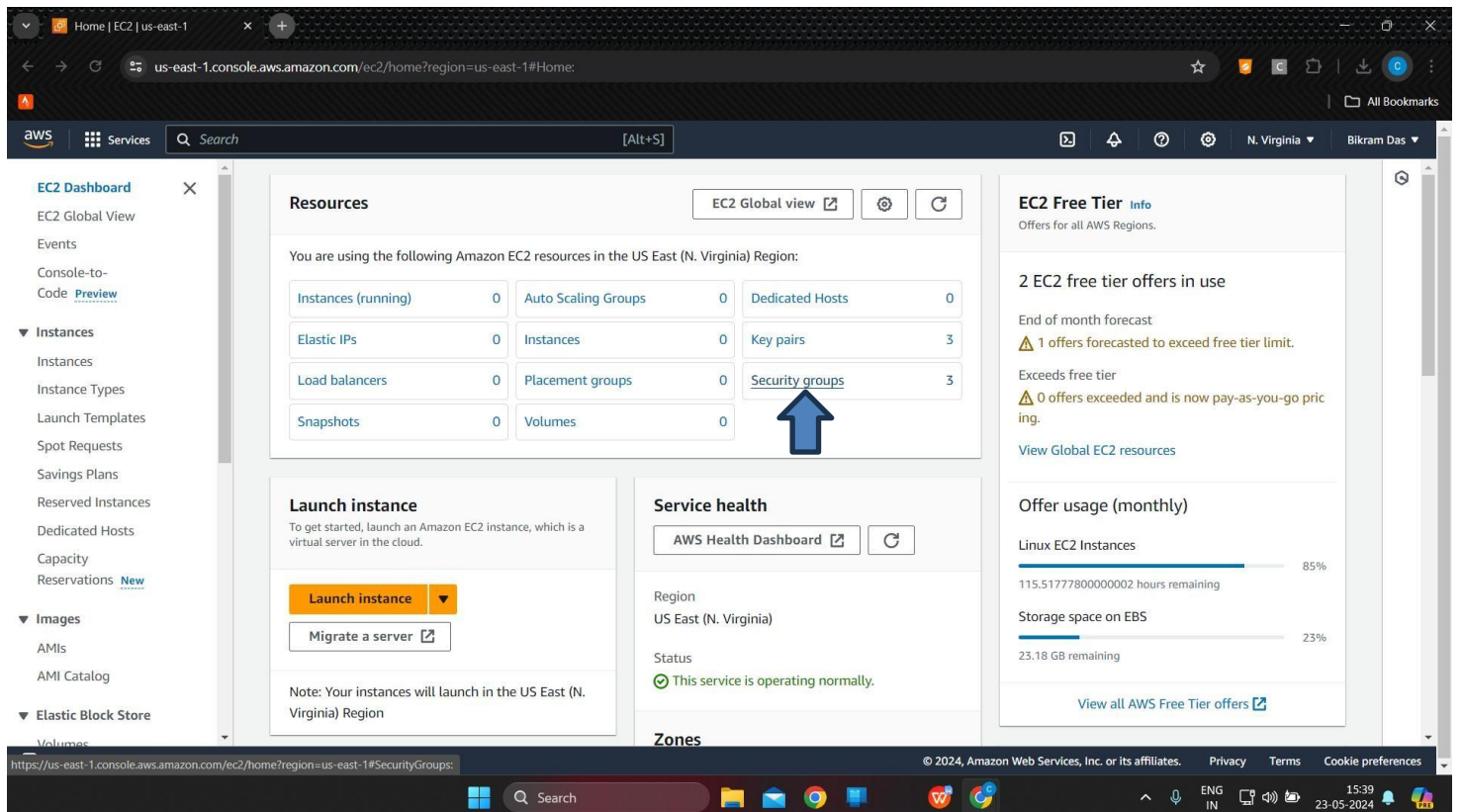
# Assignment-12

**Problem Statement:** Deploy and run the project in AWS without using port  
**Steps:-**

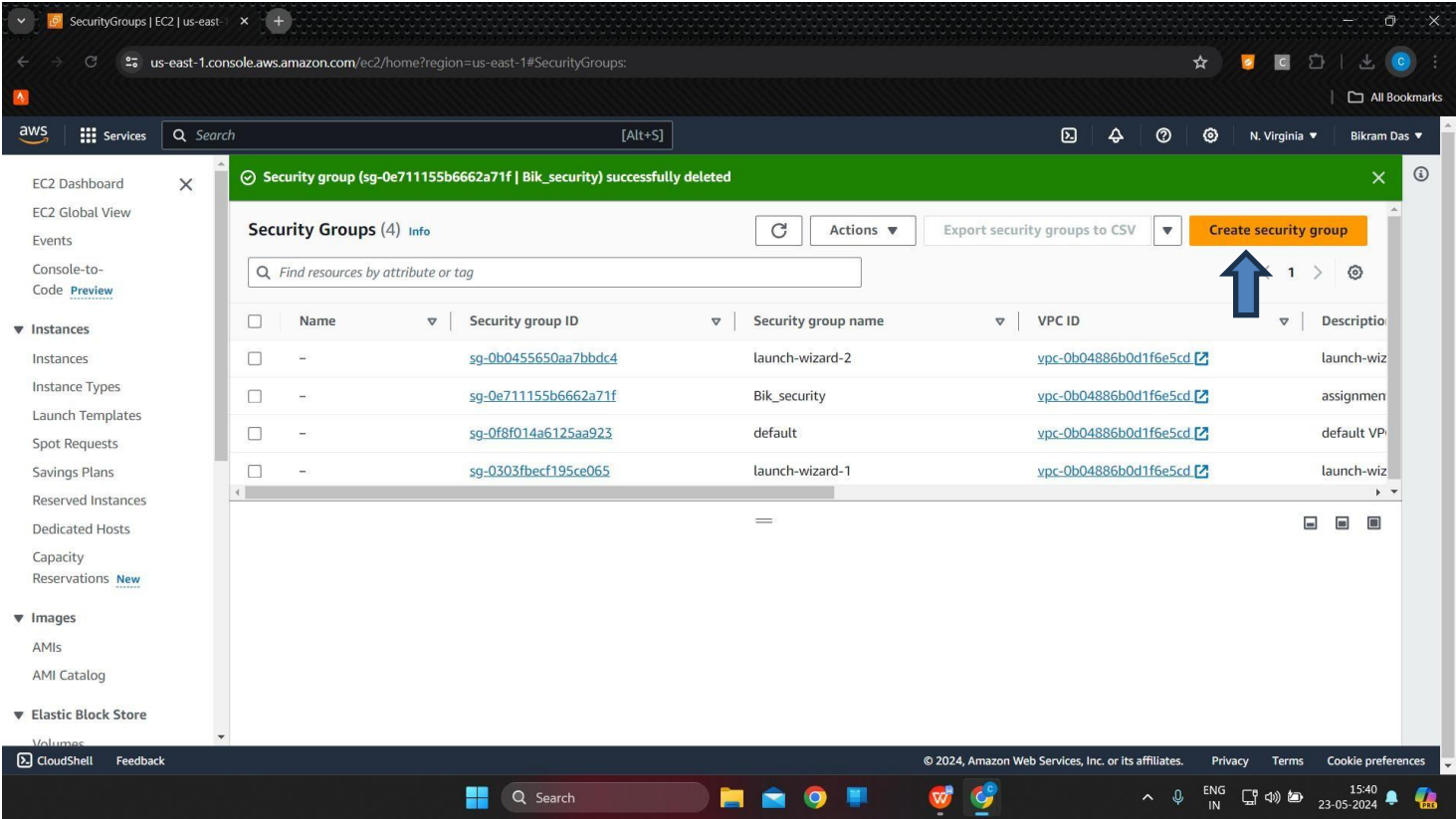
1. Sign-in to the console and Click to the EC2



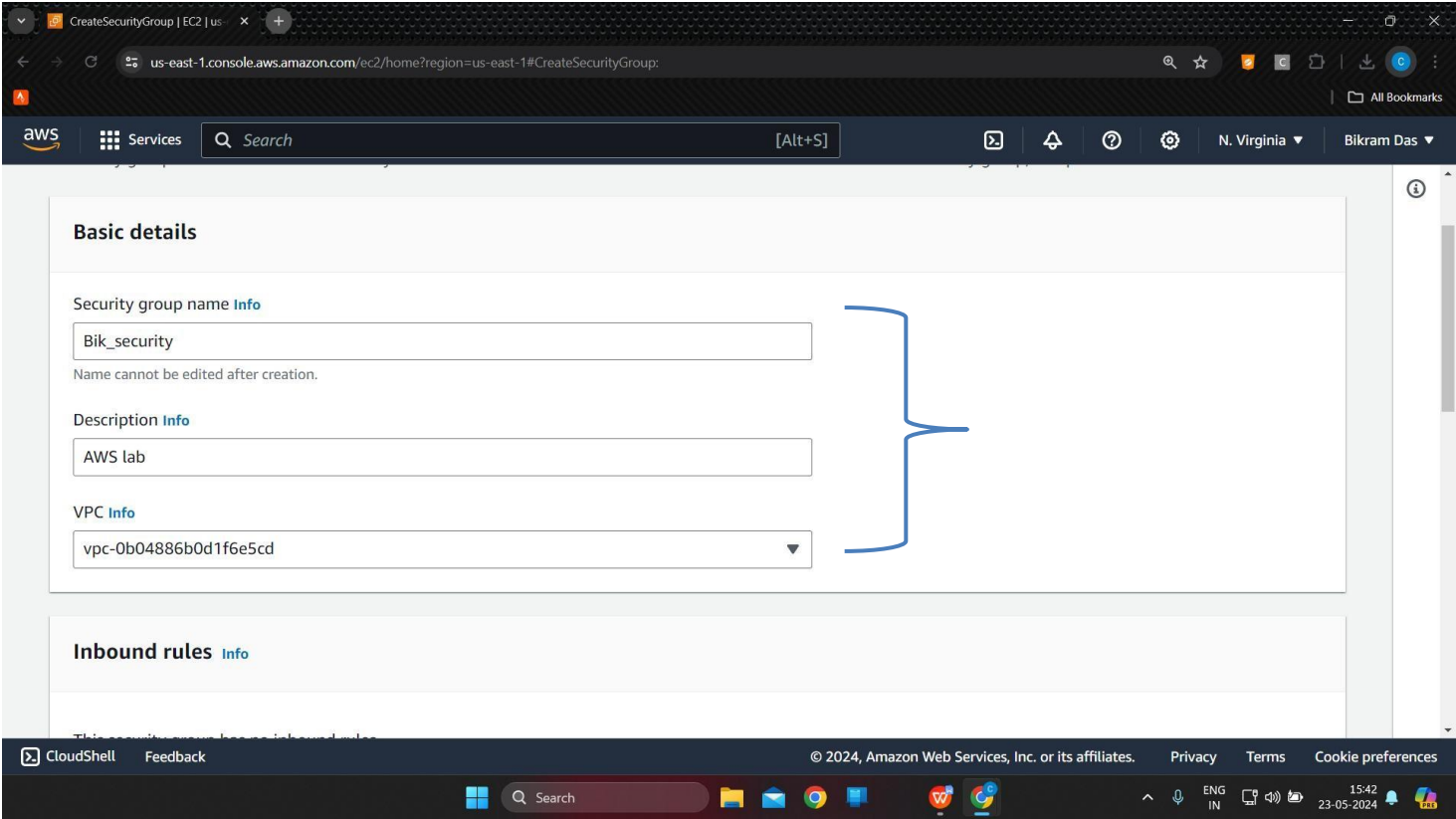
2. click on security group.



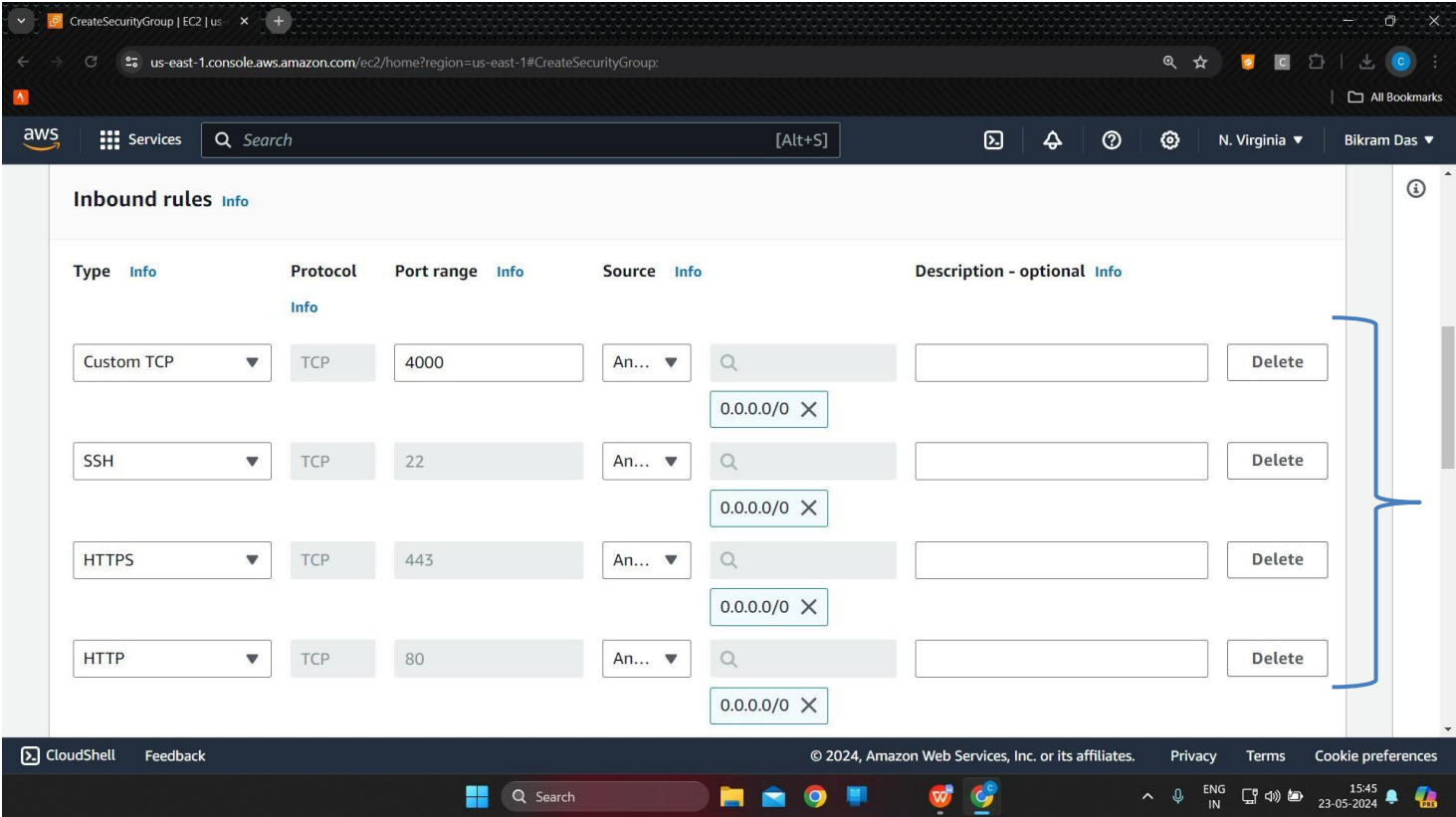
3. Click on Create Security Group



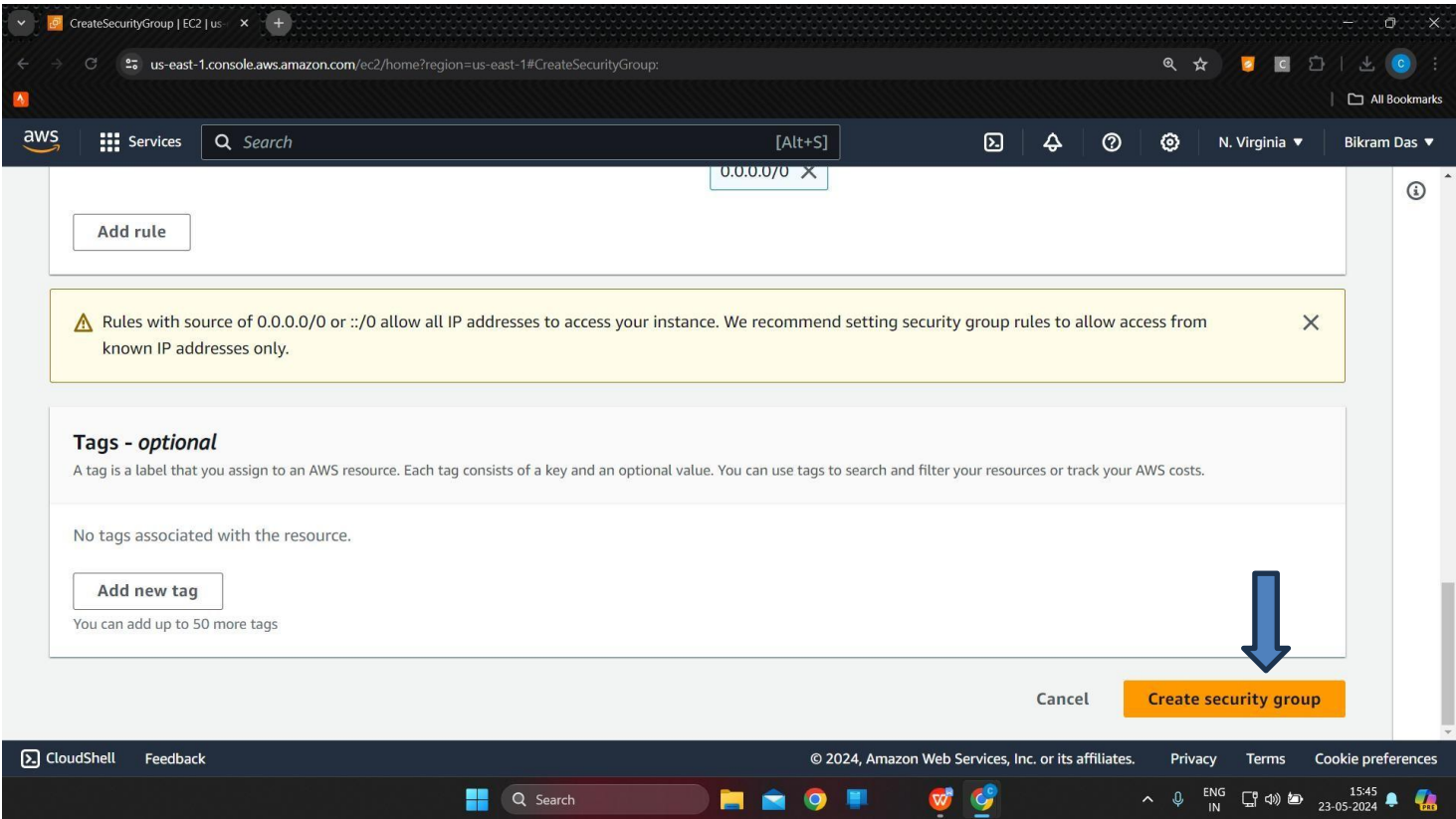
4. Now give proper name and description.



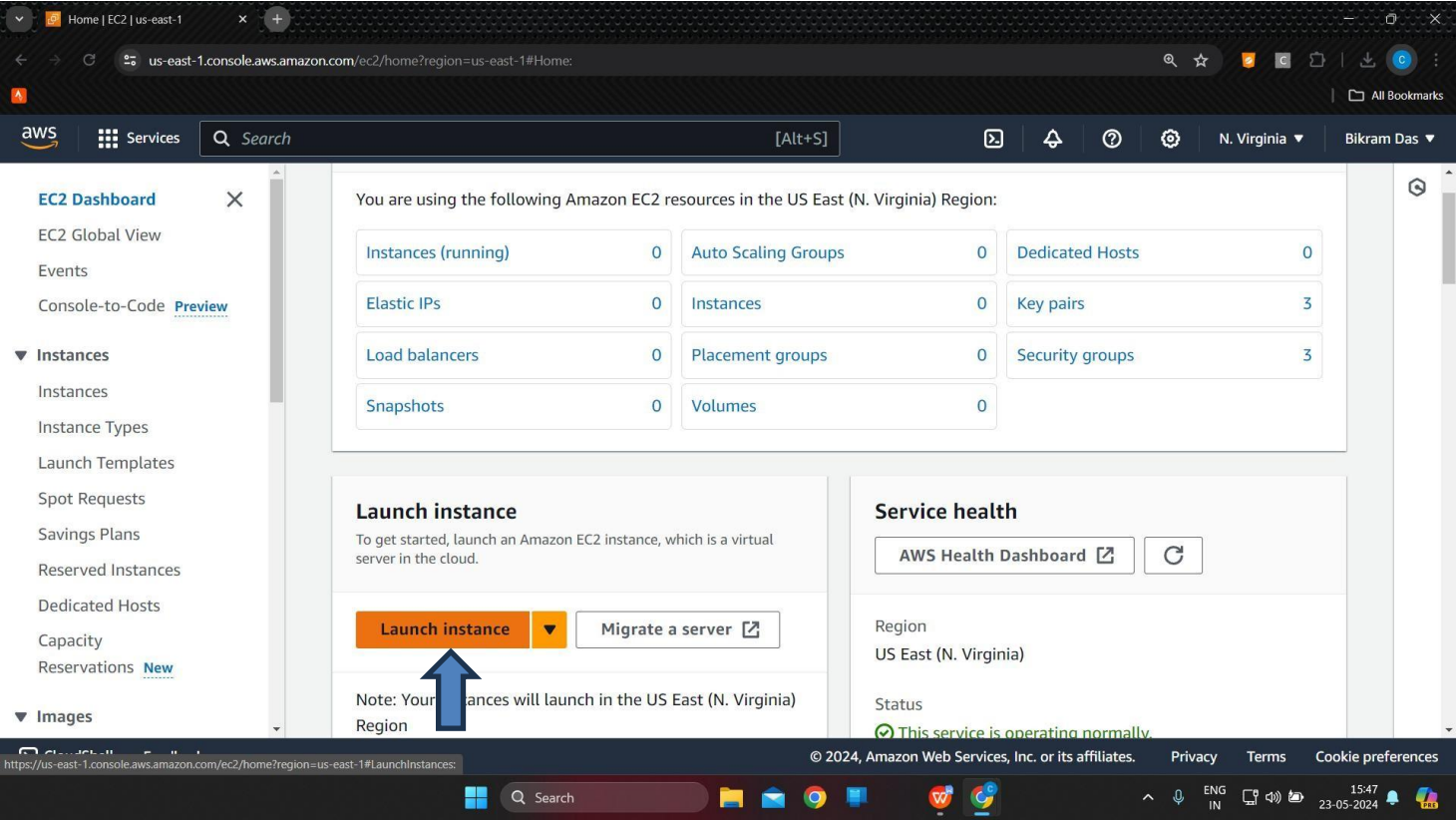
5.Now in inbound rules click on Add rule and in this way add 4 security rules of Custom TCP,SSH,HTTP,HTTPS.



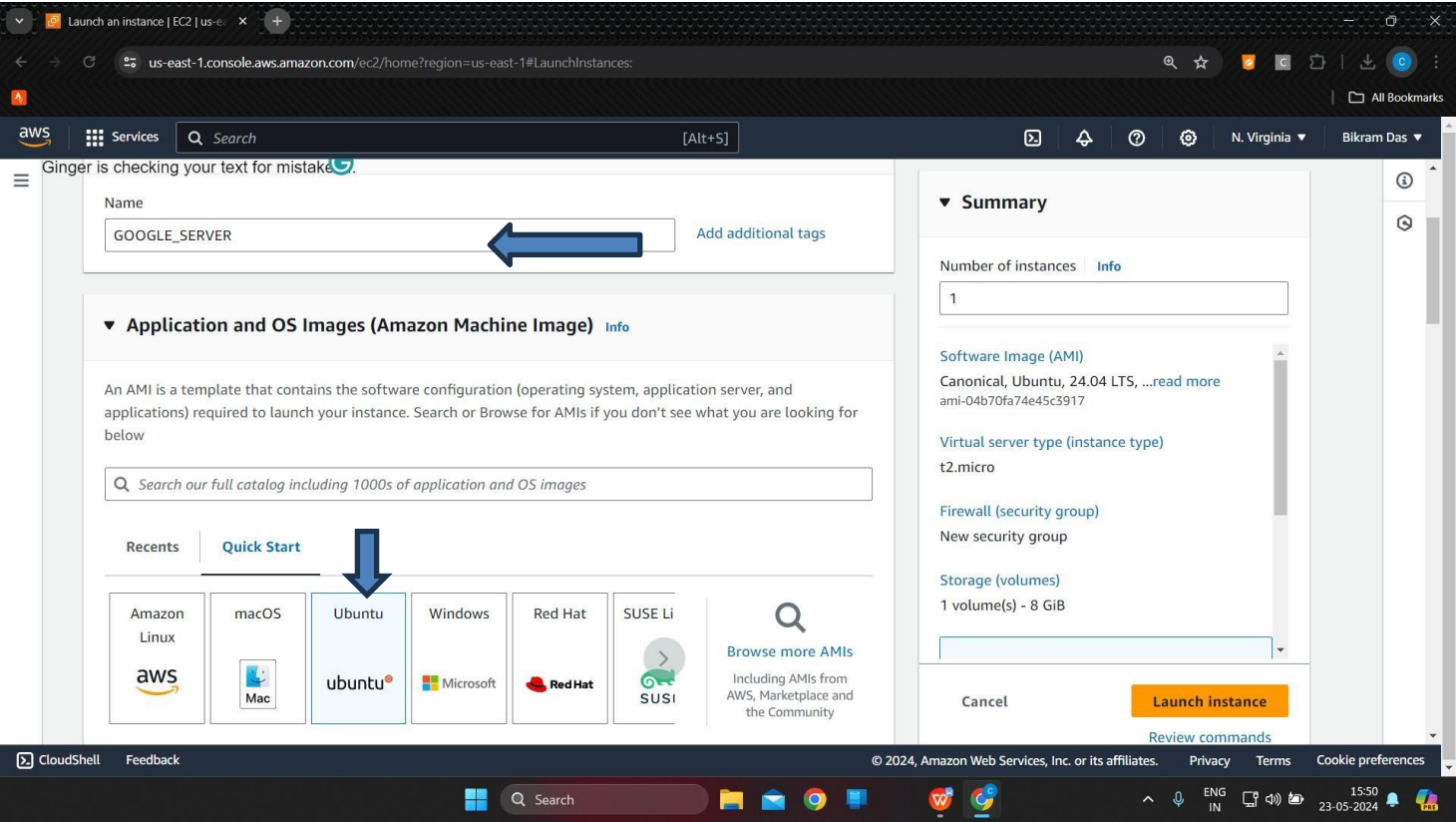
6. Now click on Create Security Group



7. Now under the EC2 Dashboard click on Launch Instance

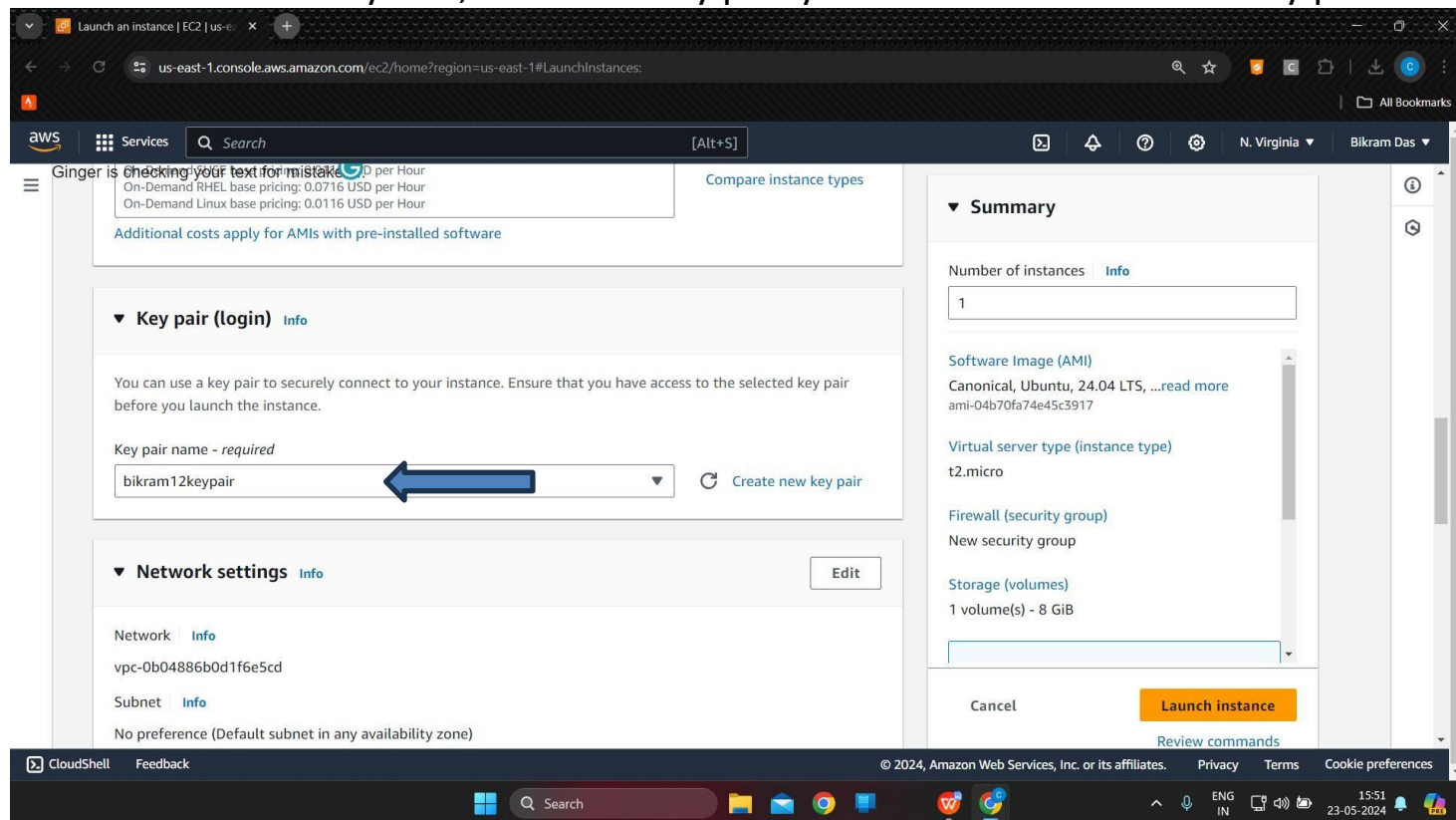


8. Give a name and click on Ubuntu under Quick Start.

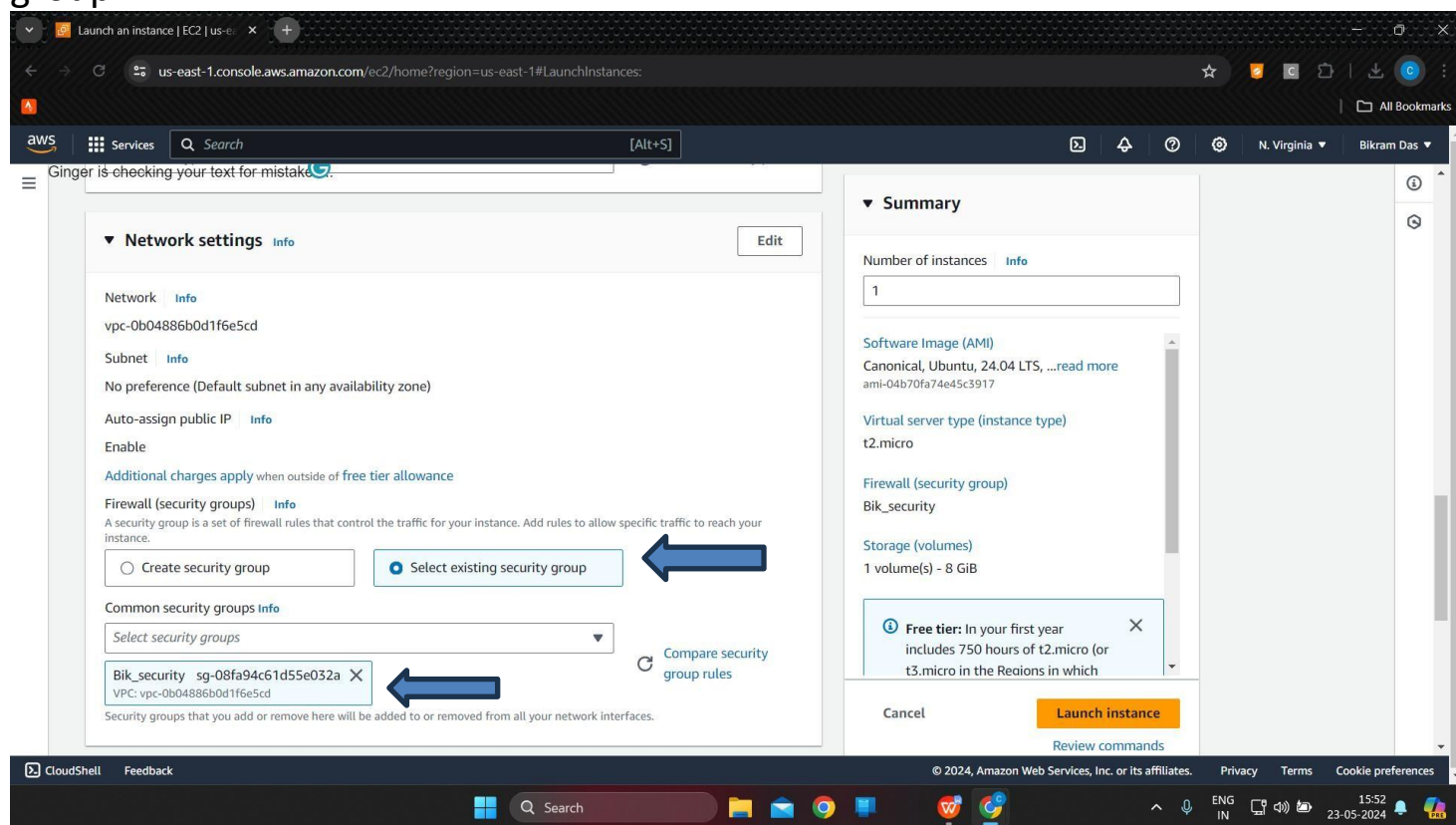




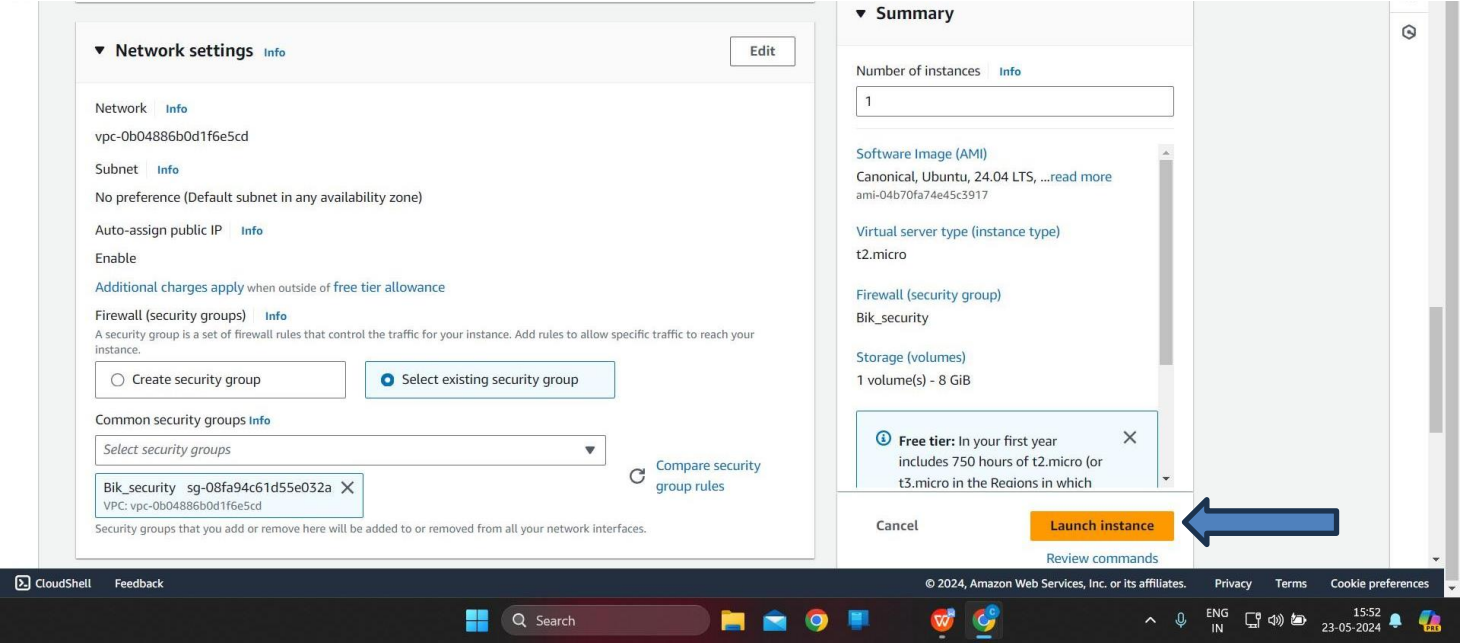
## 9. Now under the Key Pair, Select the Key pair you have or create a new key pair



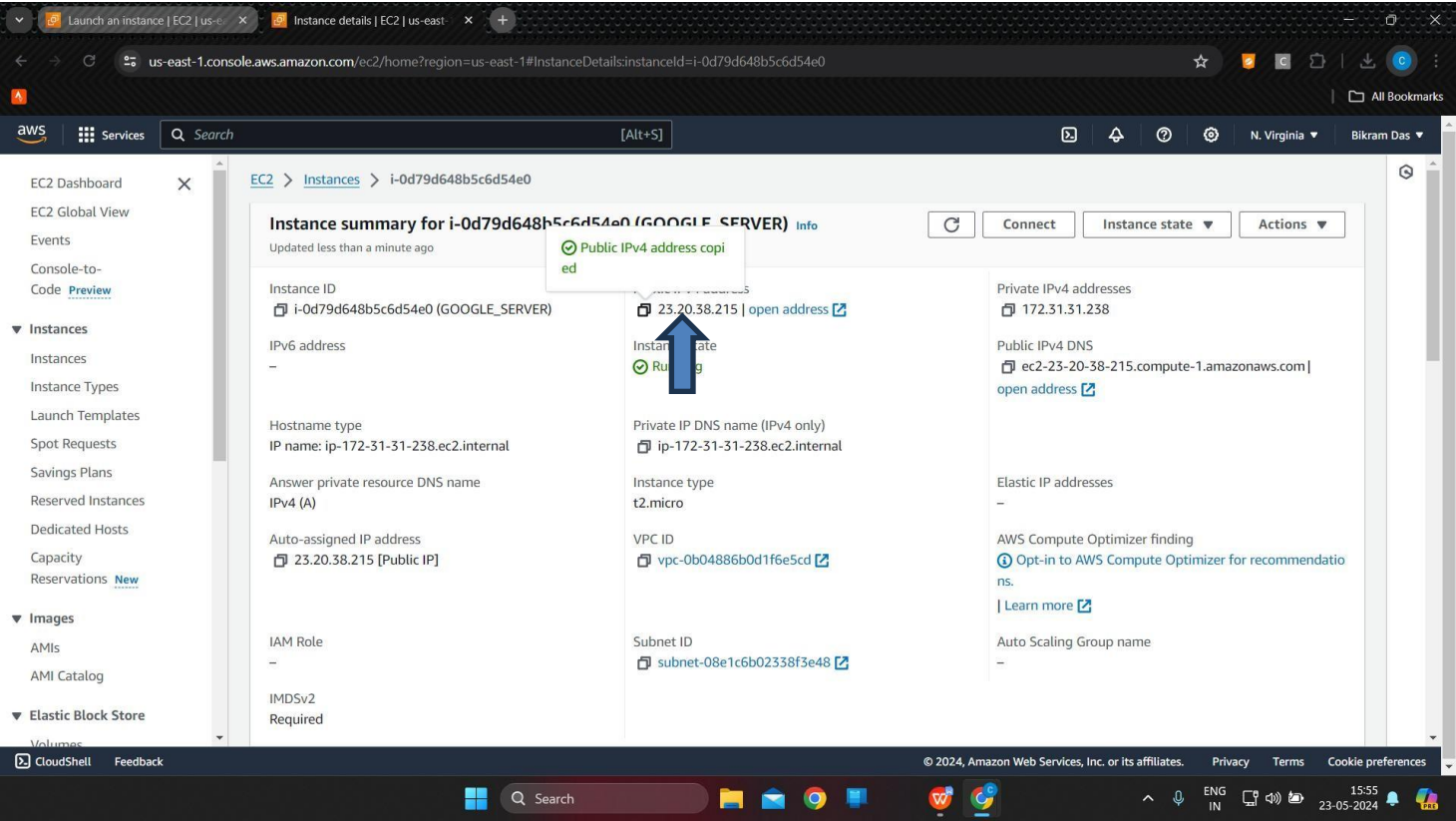
## 10. Now Navigate to the Network Settings and select "select existing security group"



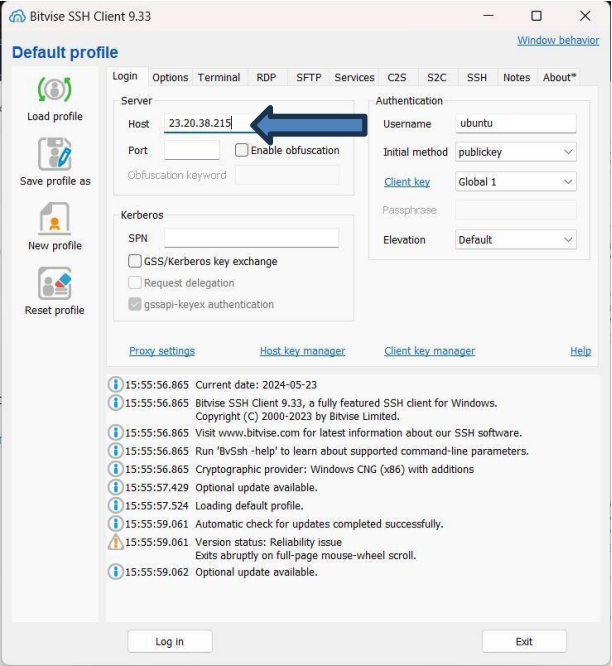
11. Now without any further changes click on the Launch Instance



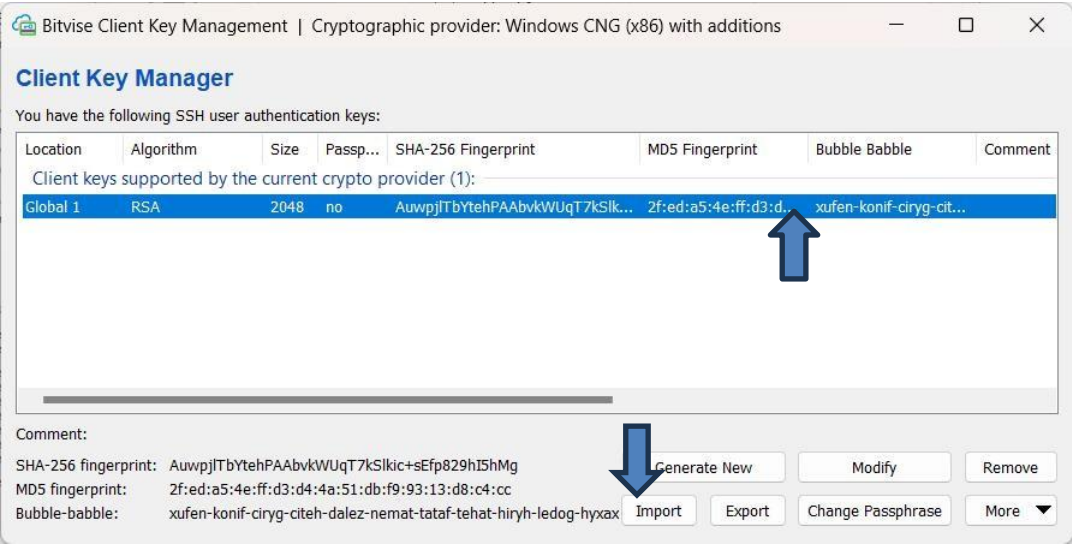
12. Click on instance and copy public IPv4 address



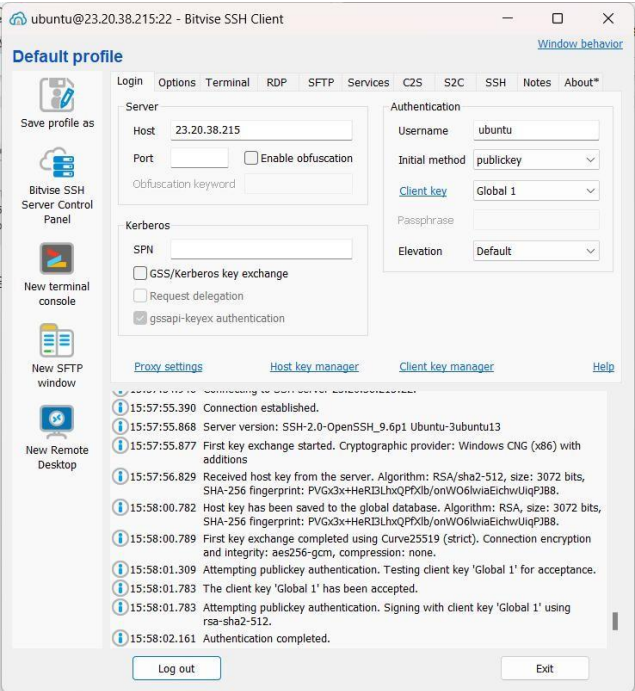
13. Paste it in host of BitVise SSH client



14. click on the client key manager and import the key.



15. click on login and choose accept and save.

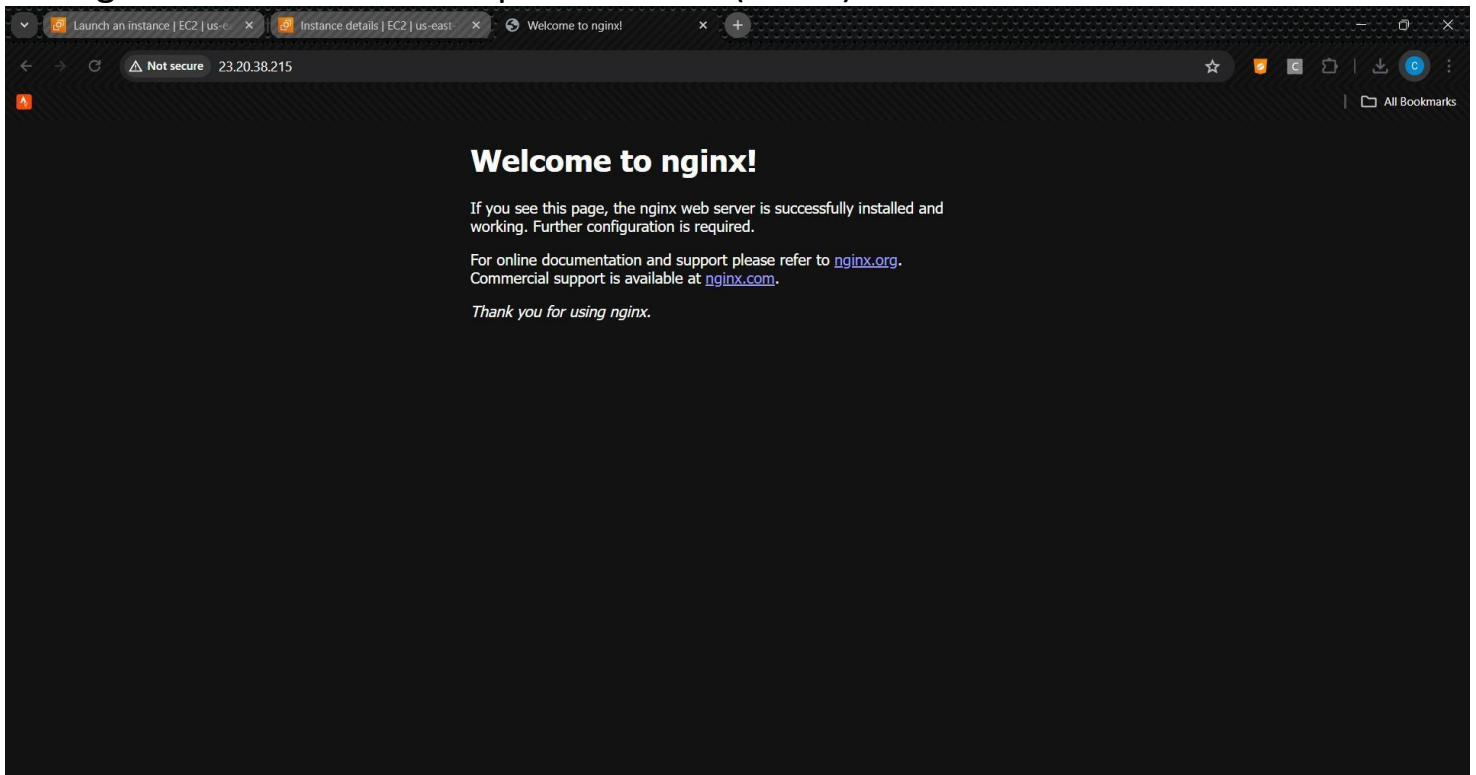


16. Now open terminal after login and then write all these commands :

- pwd
- sudo apt-get update
- sudo apt-get upgrade
- sudo apt-get install nginx
- curl -SL https://deb.nodesource.com/setup\_16.x | sudo -E bash -
- sudo apt install nodejs
- git clone [https://github.com/BikrAm2003/Moumita\\_maam\\_aws\\_repo.git](https://github.com/BikrAm2003/Moumita_maam_aws_repo.git)
- cd Moumita\_maam\_aws\_repo
- npm install
- node index.js

```
ubuntu@ip-172-31-31-238:~/Moumita_maam_aws_repo$ node index.js  
Started server
```

17. Now server has started. If we paste the ipv4 address in url section then we can see nginx has started. To stop server click (ctrl+c).



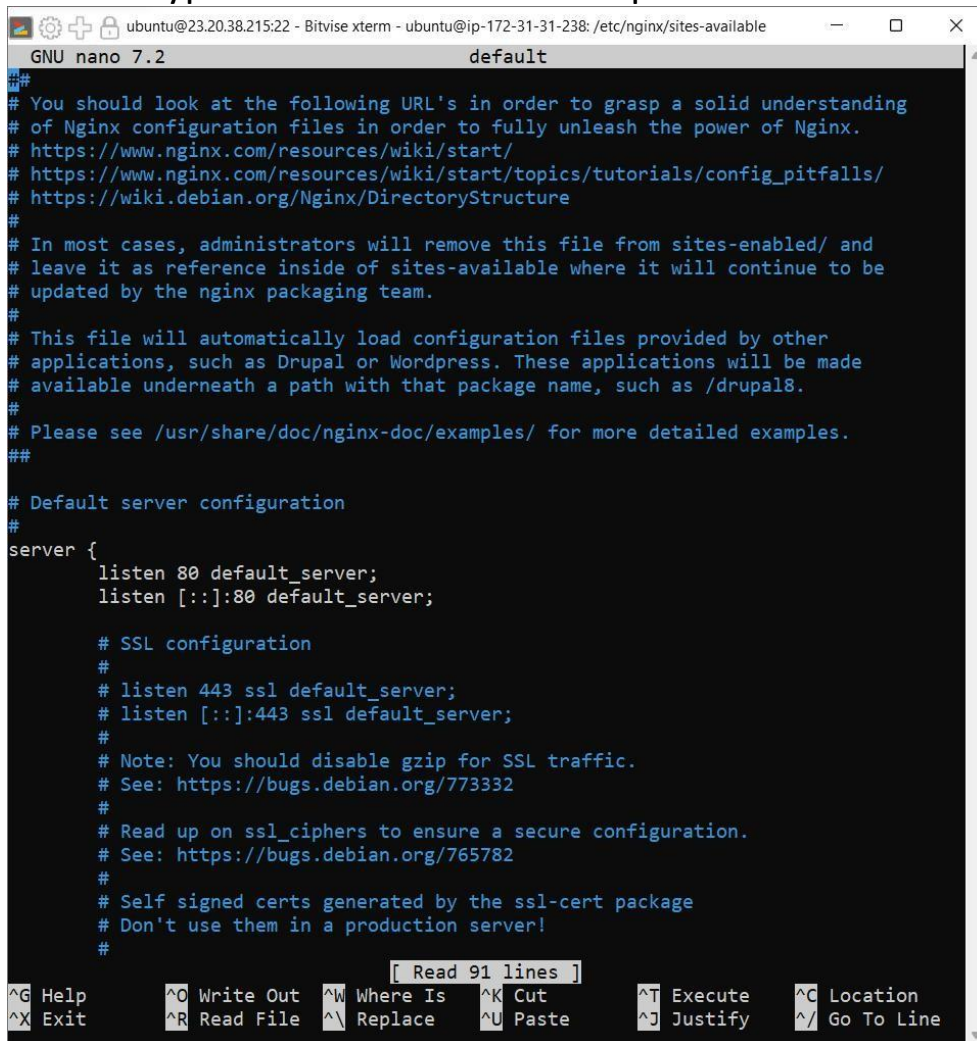


18. Now write these all commands:

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

```
ubuntu@ip-172-31-31-238:~/Moumita_maam_aws_repo$ cd /
ubuntu@ip-172-31-31-238:/$ pwd
/
ubuntu@ip-172-31-31-238:/$ cd etc/nginx/sites-available/
ubuntu@ip-172-31-31-238:/etc/nginx/sites-available$
```

19. Now type `sudo nano` default to open GNU editor.

A screenshot of a terminal window showing the nano text editor. The title bar indicates the file is 'default' in the directory '/etc/nginx/sites-available'. The editor content includes a header with links to Nginx documentation, a section for default server configuration, and a 'server' block with 'listen' directives for port 80 and SSL on port 443. The bottom status bar shows 'Read 91 lines' and various keyboard shortcuts for editor functions like Help, Write Out, Where Is, Cut, Execute, Location, Exit, Read File, Replace, Paste, Justify, and Go To Line.

```
GNU nano 7.2 default
#
# You should look at the following URL's in order to grasp a solid understanding
# of Nginx configuration files in order to fully unleash the power of Nginx.
# https://www.nginx.com/resources/wiki/start/
# https://www.nginx.com/resources/wiki/start/topics/tutorials/config_pitfalls/
# https://wiki.debian.org/Nginx/DirectoryStructure
#
# In most cases, administrators will remove this file from sites-enabled/ and
# leave it as reference inside of sites-available where it will continue to be
# updated by the nginx packaging team.
#
# This file will automatically load configuration files provided by other
# applications, such as Drupal or Wordpress. These applications will be made
# available underneath a path with that package name, such as /drupal8.
#
# Please see /usr/share/doc/nginx-doc/examples/ for more detailed examples.
##

# Default server configuration
#
server {
    listen 80 default_server;
    listen [::]:80 default_server;

    # 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!
    #

    [ Read 91 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

20. There at first go to location area and comment all codes and the 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;
}
```

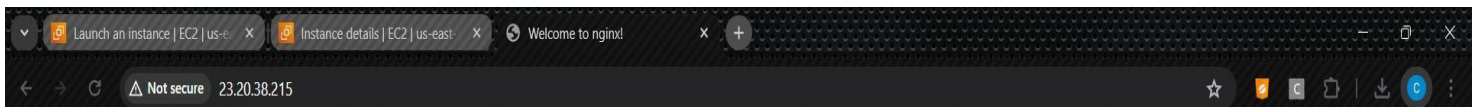
```
location / {  
    # First attempt to serve request as file, then  
    # as directory, then fall back to displaying a 404.  
    proxy_pass http://localhost:4000;  
    proxy_http_version 1.1;  
    proxy_set_header Upgrade $http_upgrade;  
    proxy_set_header Connection 'upgrade';  
    proxy_set_header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

21. After this click ctrl+x, then y then click enter

22. Now open new terminal and write `cd Moumita_maam_aws_repo`.

23. Write `sudo systemctl restart nginx`.

24. 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.



Hello mckv