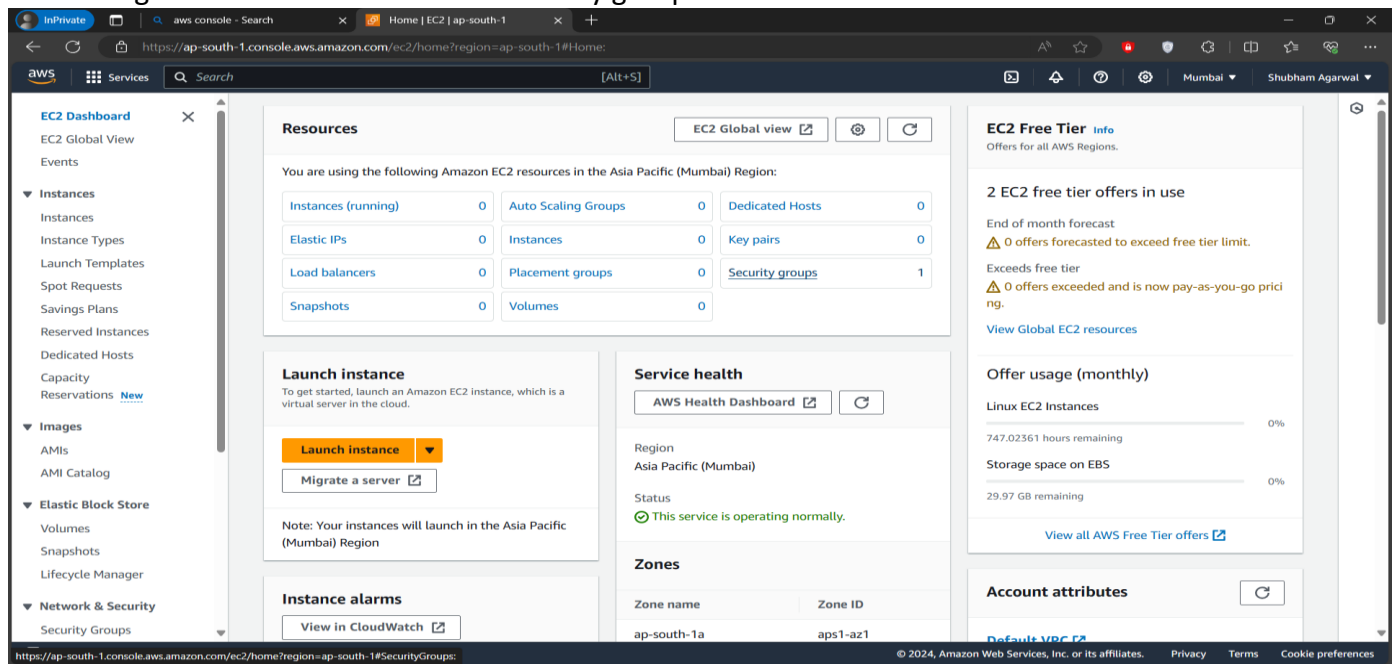


## Assignment No : 12

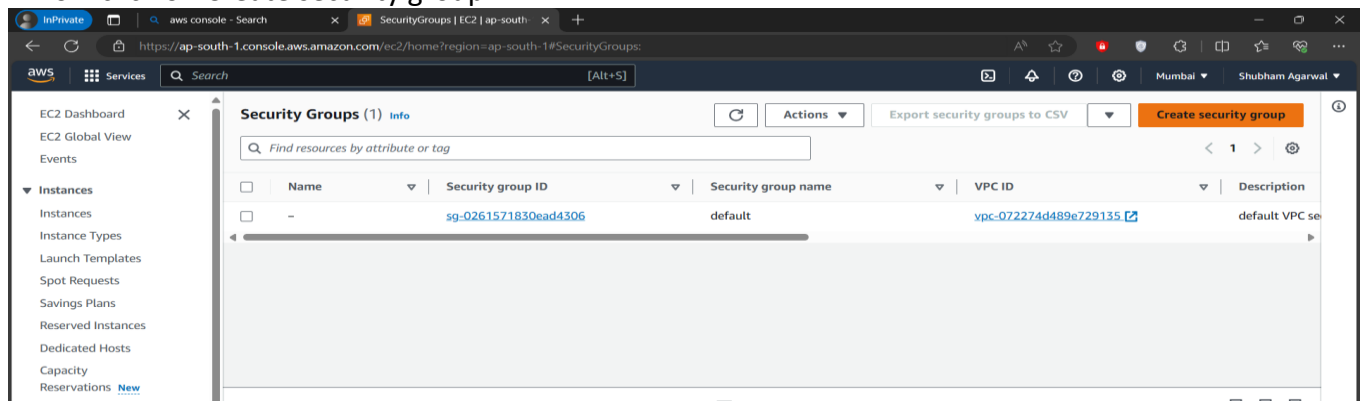
**Problem Statement :** Deploy and run the project in AWS without using port.

### Steps:

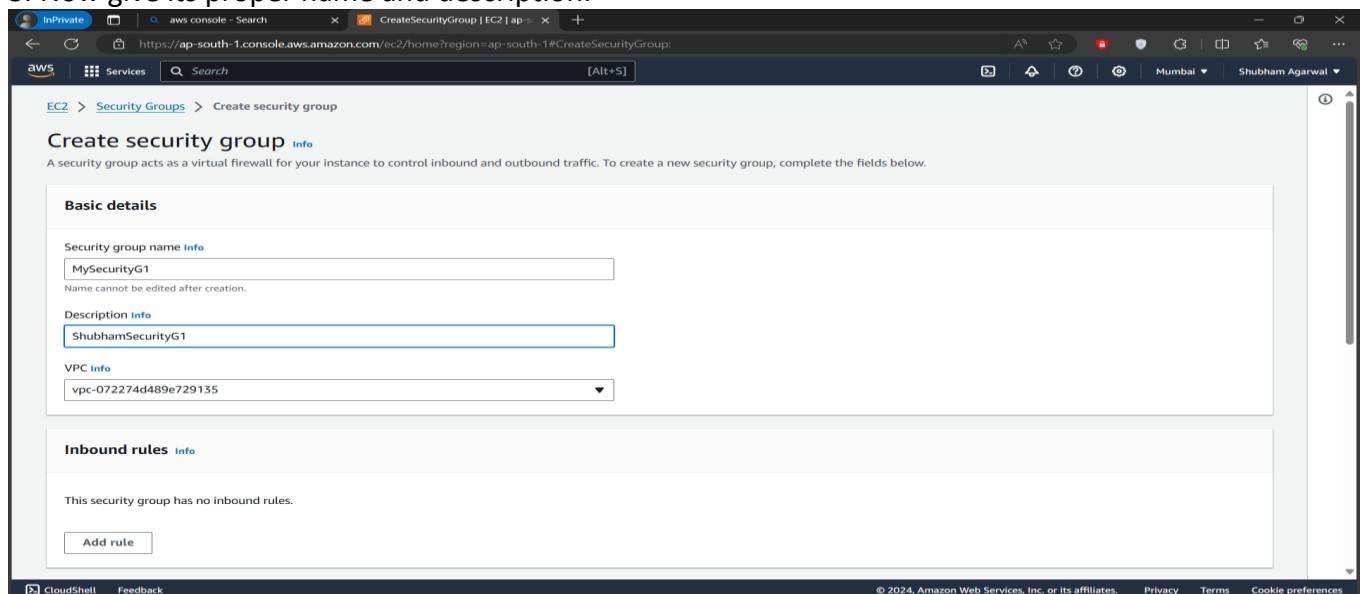
1. At first go to EC2 and then click on Security group.



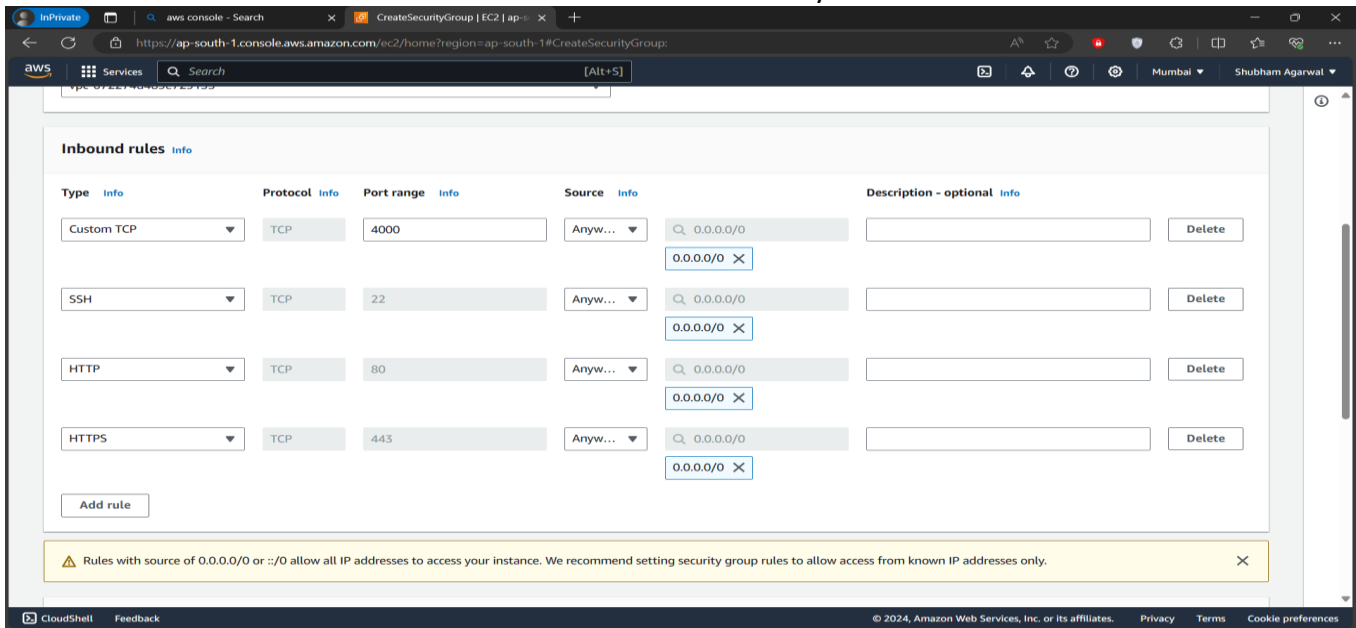
2. Now click on Create security group.



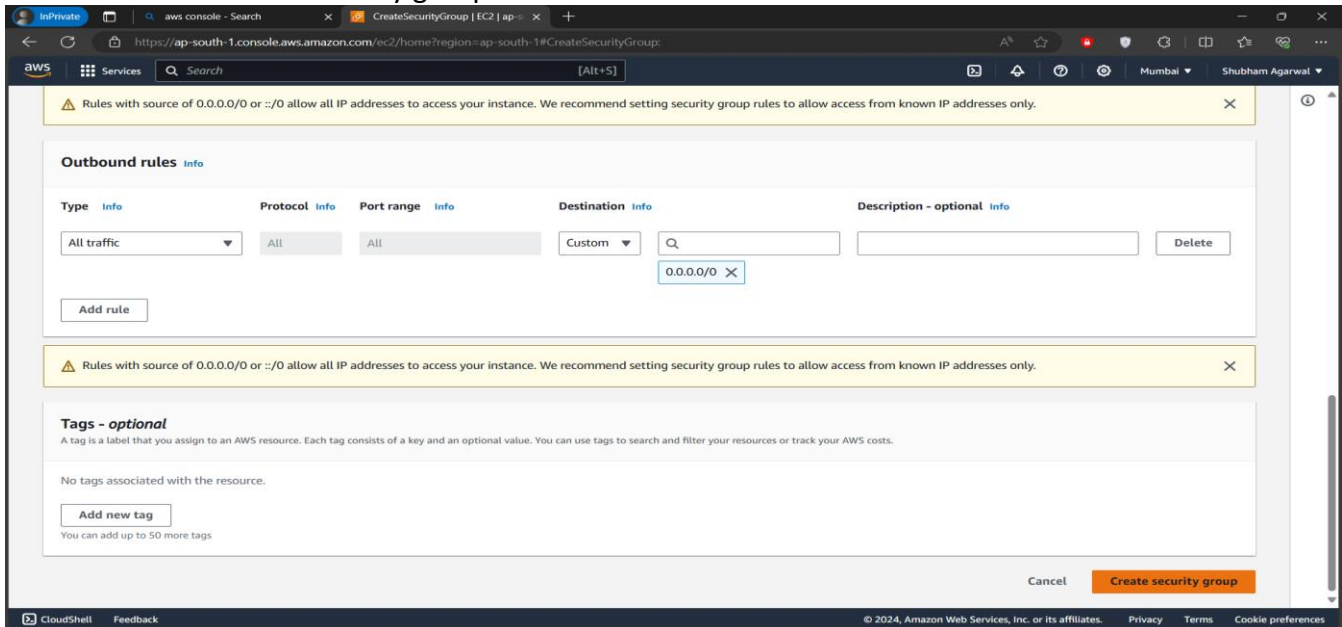
3. Now give its proper name and description.



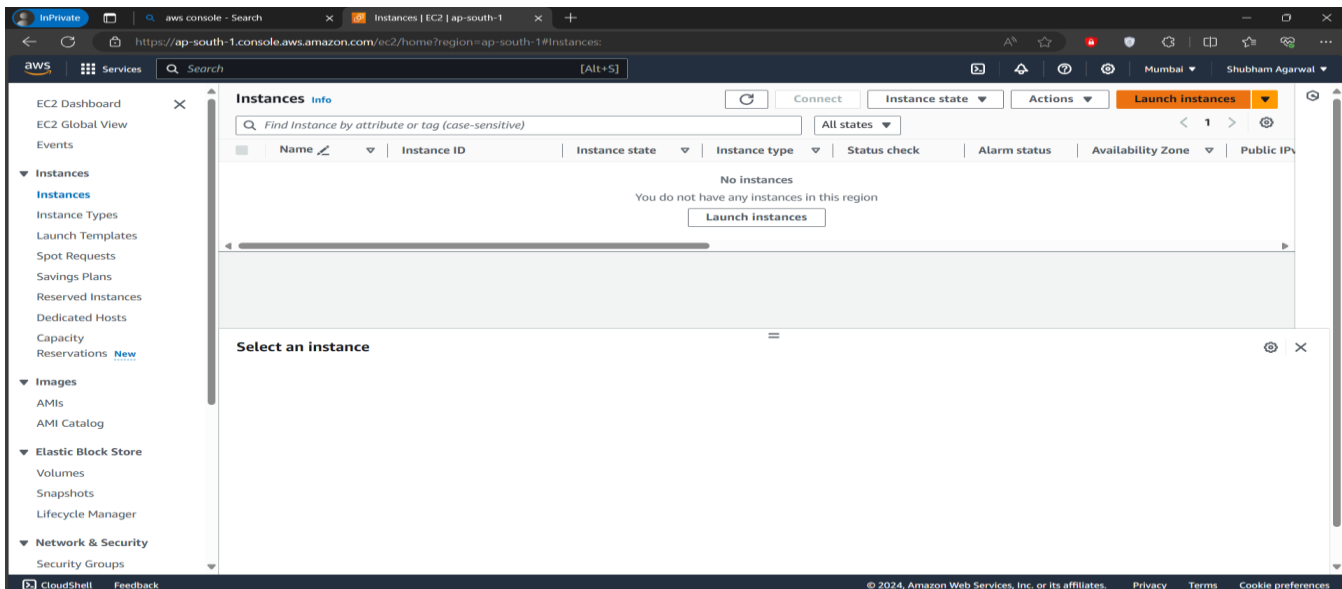
4. Now in inbound click on Add rule and add these four security rules.



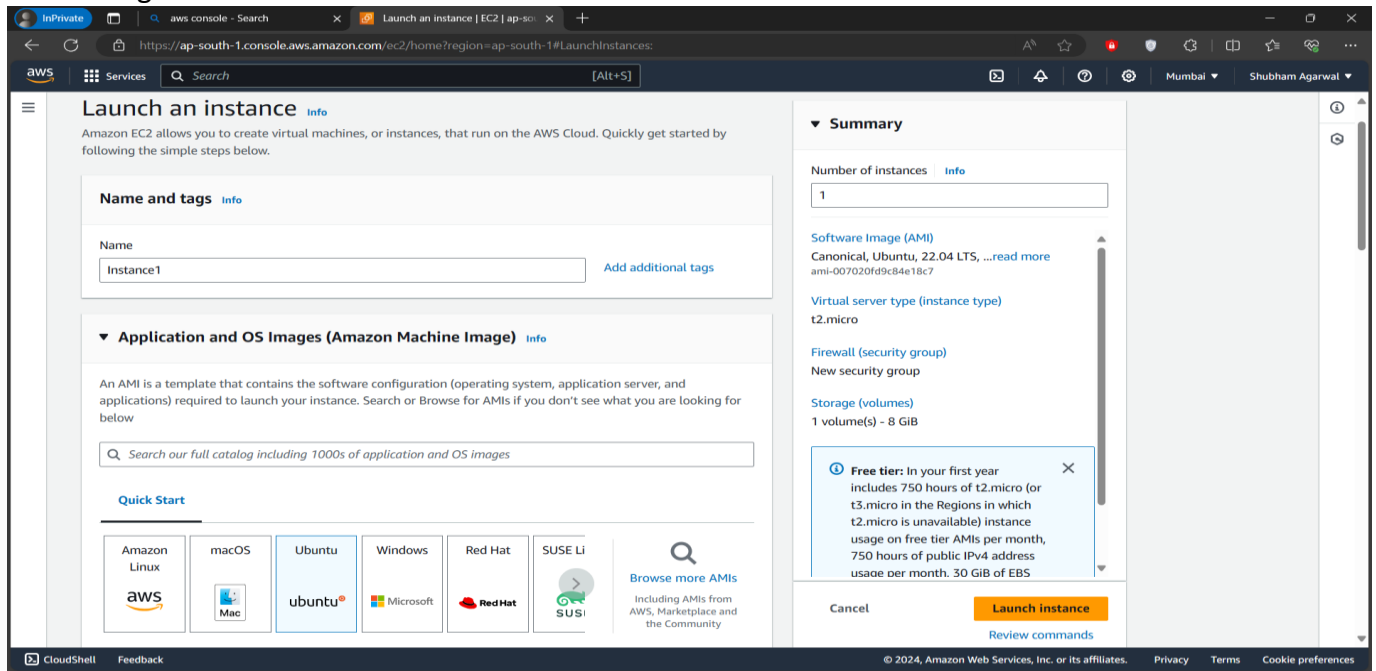
5. Now click on Create Security group.



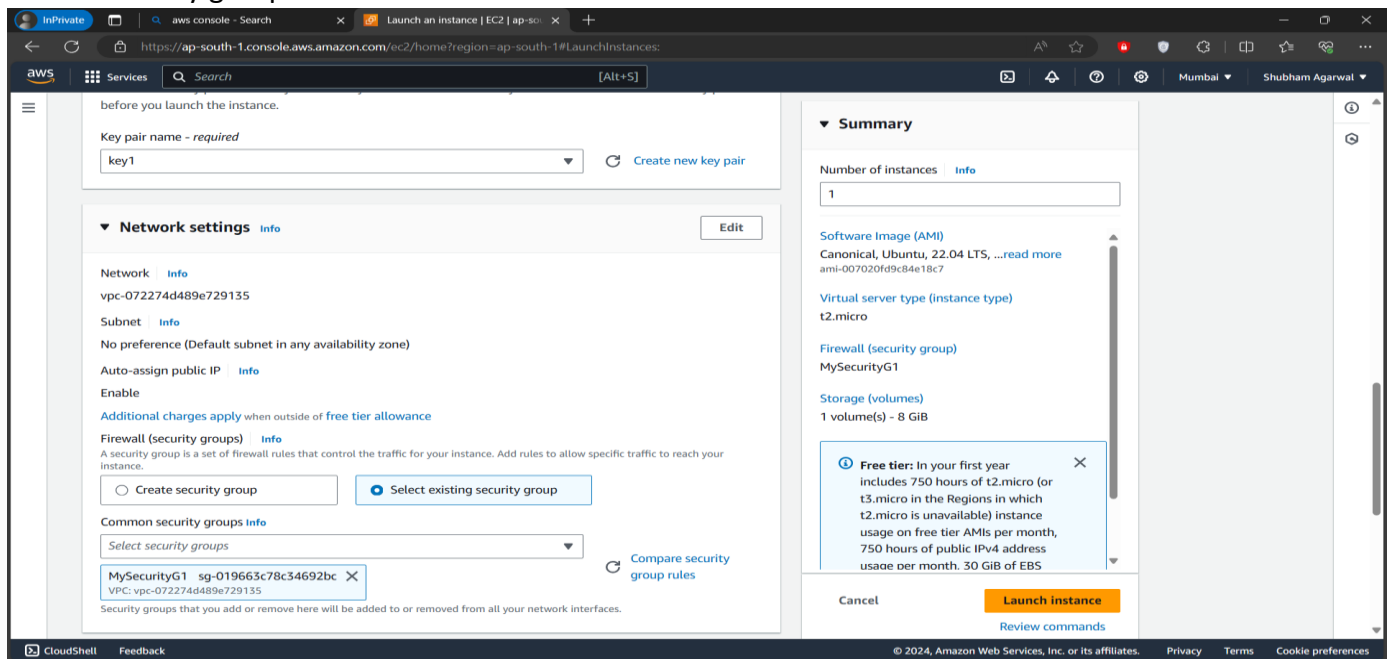
6. Now Go to Instances and then Launch Instance.



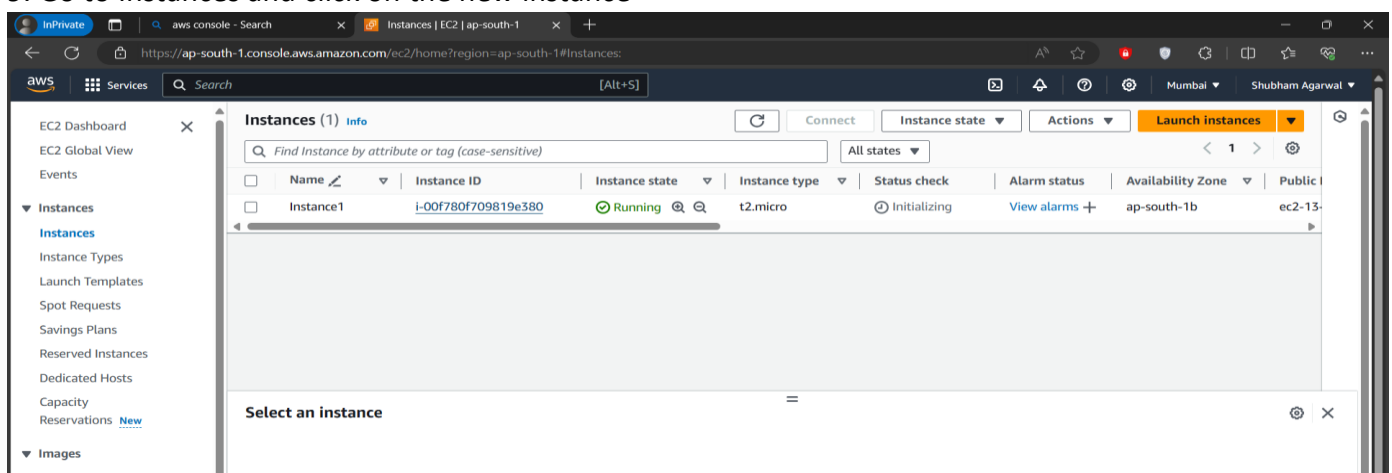
## 7. After it give name and then click on Ubuntu.



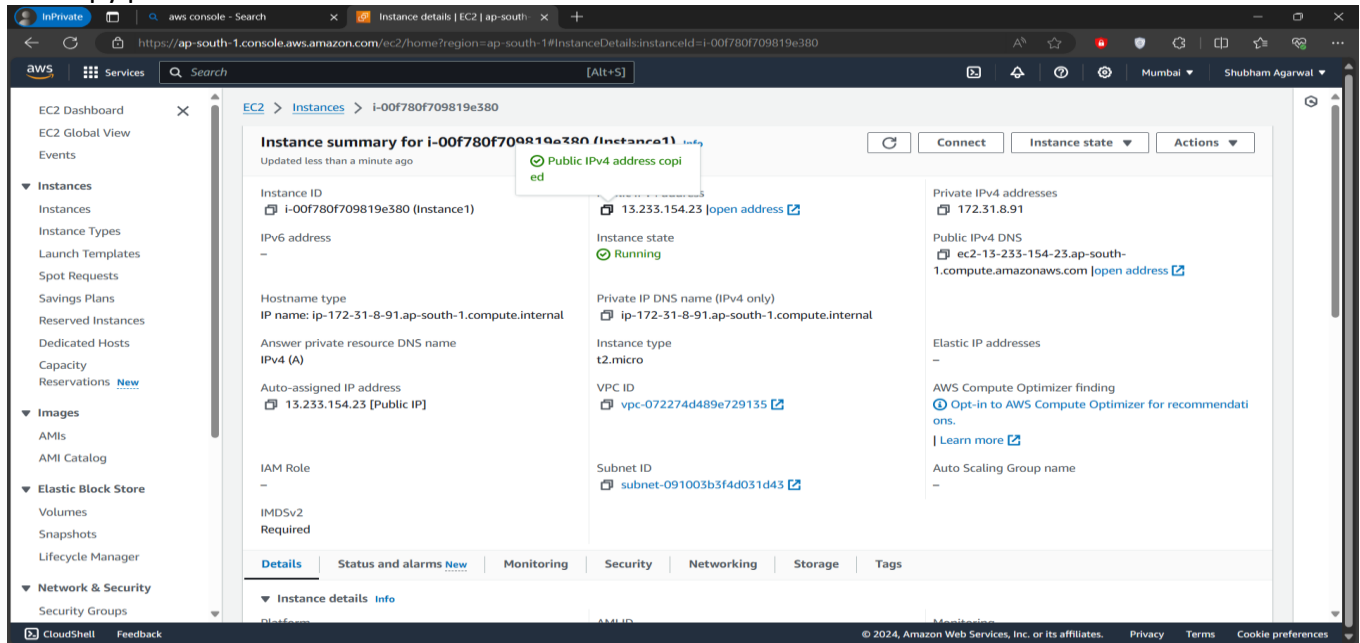
## 8. Now select key pair and also then click on Select existing security group and select the new security group created then click on Launch Instance.



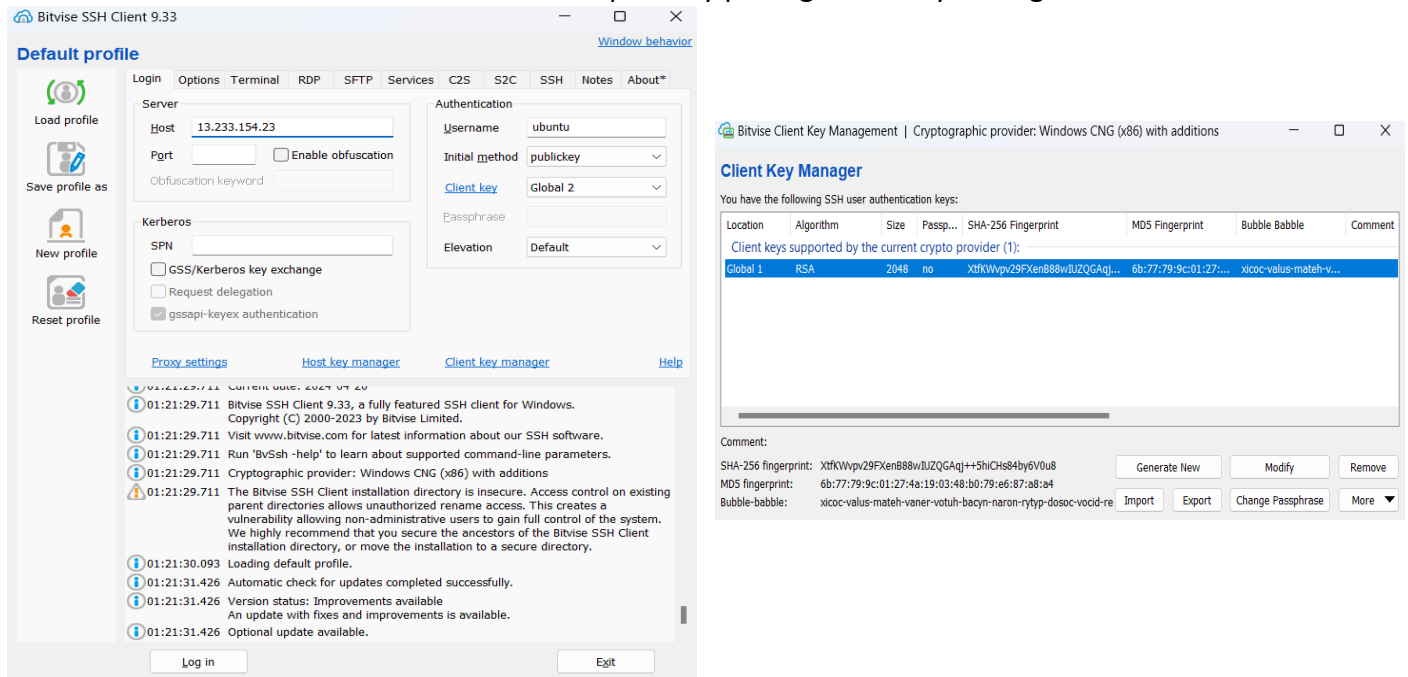
## 9. Go to Instances and click on the new instance



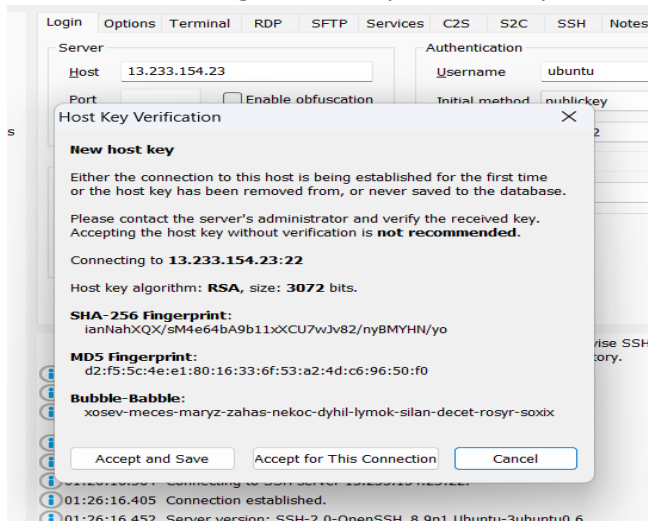
## 10. Copy public IPv4 address.



## 11. Paste it in host of Bitwise SSH Client and import key pairing Client key manager.



## 12. Now click on LogIn and verify the host key and click on save



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

- Pwd
- sudo apt-get update

```
ubuntu@13.233.154.23:22 - Bitvise xterm - ubuntu@ip-172-31-8-91: ~
ubuntu@ip-172-31-8-91:~$ pwd
/home/ubuntu
ubuntu@ip-172-31-8-91:~$ sudo apt-get update
Hit:1 http://ap-south-1-ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://ap-south-1-ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://ap-south-1-ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Get:4 http://ap-south-1-ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:5 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:6 http://ap-south-1-ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]
Get:7 http://ap-south-1-ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]
Get:8 http://ap-south-1-ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [217 kB]
Get:9 http://ap-south-1-ec2.archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]
Get:10 http://ap-south-1-ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [8372 B]
```

- sudo apt-get upgrade

```
ubuntu@13.233.154.23:22 - Bitvise xterm - ubuntu@ip-172-31-8-91: ~
Reading package lists... Done
ubuntu@ip-172-31-8-91:~$ 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-110n update-manager-core
The following packages will be upgraded:
  apt apt-utils bash bsdxtractils bsdtails 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 snapd update-notifier-common util-linux uuid-runtime vim vim-common vim-runtime vim-tiny xxd
45 upgraded, 0 newly installed, 0 to remove and 7 not upgraded.
Need to get 56.0 MB of archives.
```

- sudo apt-get install nginx

```
ubuntu@13.233.154.23:22 - Bitvise xterm - ubuntu@ip-172-31-8-91: ~
Try: sudo apt install <deb name>
ubuntu@ip-172-31-8-91:~$ 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 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 nginx-common nginx-core
```

- curl -fsSL https://deb.nodesource.com/setup\_16.x | bash -

```
ubuntu@13.233.154.23:22 - Bitvise xterm - ubuntu@ip-172-31-8-91: ~
rmat?)
curl: (6) Could not resolve host: sudo
ubuntu@ip-172-31-8-91:~$ curl -fsSL https://deb.nodesource.com/setup_16.x | sudo -E bash -

=====
DEPRECATION WARNING
=====
Node.js 16.x is no longer actively supported!

You will not receive security or critical stability updates for this version.

You should migrate to a supported version of Node.js as soon as possible.
Use the installation script that corresponds to the version of Node.js you
wish to install. e.g.

* https://deb.nodesource.com/setup_16.x - Node.js 16 "Gallium" (deprecated)
* https://deb.nodesource.com/setup_18.x - Node.js 18 "Hydrogen" (Maintenance)
* https://deb.nodesource.com/setup_20.x - Node.js 20 LTS "Iron" (recommended)
* https://deb.nodesource.com/setup_21.x - Node.js 21 "Iron" (current)

Please see https://github.com/nodejs/Release for details about which
version may be appropriate for you.

The NodeSource Node.js distributions repository contains
information both about supported versions of Node.js and supported Linux
distributions. To learn more about usage, see the repository:
https://github.com/node-source/distributions

=====
Continuing in 10 seconds ...
2024-04-19 20:13:02 - Installing pre-requisites
```

- `sudo apt install nodejs`

```
ubuntu@13.233.154.23:22 - Bitvise xterm - ubuntu@ip-172-31-8-91: ~
.22.9~dfsg-1ubuntu3.5 [2411 kB]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 nodejs amd64 12.2
2.9~dfsg-1ubuntu3.5 [122 kB]
Fetched 13.7 MB in 0s (48.7 MB/s)
Selecting previously unselected package javascript-common.
(Reading database ... 65516 files and directories currently installed.)
Preparing to unpack .../0-javascript-common_11+nmu1_all.deb ...
Unpacking javascript-common (11+nmu1) ...
Selecting previously unselected package libjs-highlight.js.
Preparing to unpack .../1-libjs-highlight.js_9.18.5+dfsg1-1_all.deb ...
Unpacking libjs-highlight.js (9.18.5+dfsg1-1) ...
Selecting previously unselected package libc-ares2:amd64.
Preparing to unpack .../2-libc-ares2_1.18.1-1ubuntu0.22.04.3_amd64.deb ...
```

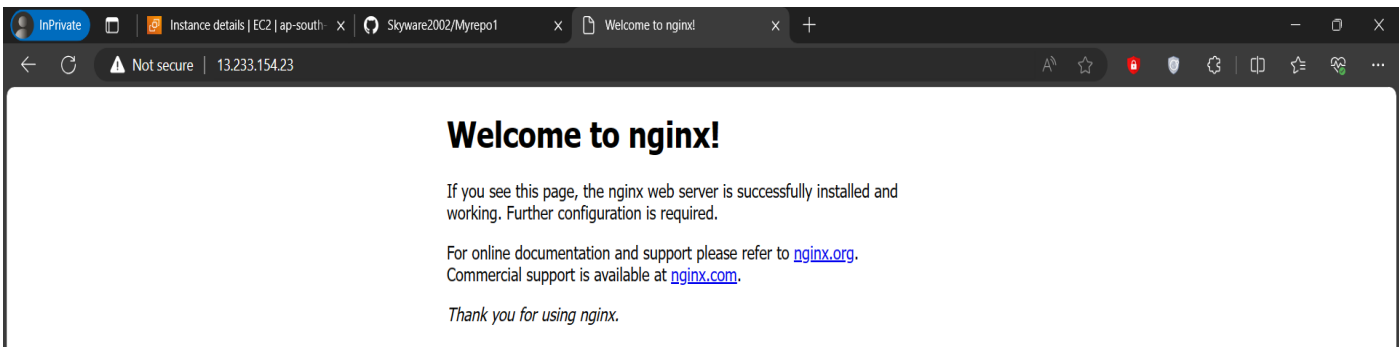
- `git clone https://github.com/Skyware2002/Myrepo1.git`
- `cd Myrepo1`

```
ubuntu@13.233.154.23:22 - Bitvise xterm - ubuntu@ip-172-31-8-91: ~/Myrepo1
No containers need to be restarted.
No user sessions are running outdated binaries.
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-8-91:~/Myrepo1$ git clone https://github.com/Skyware2002/Myrepo1.git
Cloning into 'Myrepo1'...
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 8 (delta 1), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (8/8), 49.50 KiB | 4.13 MiB/s, done.
Resolving deltas: 100% (1/1), done.
ubuntu@ip-172-31-8-91:~/Myrepo1$ cd Myrepo1
ubuntu@ip-172-31-8-91:~/Myrepo1$
```

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

```
ubuntu@13.233.154.23:22 - Bitvise xterm - ubuntu@ip-172-31-8-91: ~/Myrepo1
ubuntu@ip-172-31-8-91:~/Myrepo1$ node index.js
Started server
```

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



14. Now write these all commands:

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

```
ubuntu@13.233.154.23:22 - Bitvise xterm - ubuntu@ip-172-31-8-91: /etc/nginx/sites-available
Started server
^C
ubuntu@ip-172-31-8-91:~/Myrepo1$ cd /
ubuntu@ip-172-31-8-91:/$ pwd
/
ubuntu@ip-172-31-8-91:/$ cd etc/nginx/sites-available/
ubuntu@ip-172-31-8-91:/etc/nginx/sites-available$
```



- sudo nano default

```

GNU nano 6.2                                     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!
    #
    # include snippets/snakeoil.conf;

    root /var/www/html;
}

```

15. A new window will be opened. 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;

```

GNU nano 6.2                                     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;
}

```

16. After it click on 'ctrl + x' , then y then click enter.

```
# deny access to .htaccess files, if Apache's document root  
# concurs with nginx's one  
#  
#location ~ /\.ht {  
#    deny all;  
#}  
Save modified buffer?  
Y Yes  
N No      ^C Cancel
```

```
Started server  
^C  
ubuntu@ip-172-31-8-91:~/Myrepo1$ cd /  
ubuntu@ip-172-31-8-91:/ $ pwd  
/  
ubuntu@ip-172-31-8-91:/ $ cd etc/nginx/sites-available/  
ubuntu@ip-172-31-8-91:/etc/nginx/sites-available$ sudo nano default  
ubuntu@ip-172-31-8-91:/etc/nginx/sites-available$ cd /
```

17. Now open new terminal and write

- > cd Myrepo1
- > sudo systemctl restart nginx.
- > node index.js

```
Last login: Fri Apr 19 21:10:54 2024 from 49.37.8.11  
ubuntu@ip-172-31-8-91:~$ cd Myrepo1  
ubuntu@ip-172-31-8-91:~/Myrepo1$ sudo systemctl restart nginx  
ubuntu@ip-172-31-8-91:~/Myrepo1$ node index.js  
Started server
```

18. 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.

