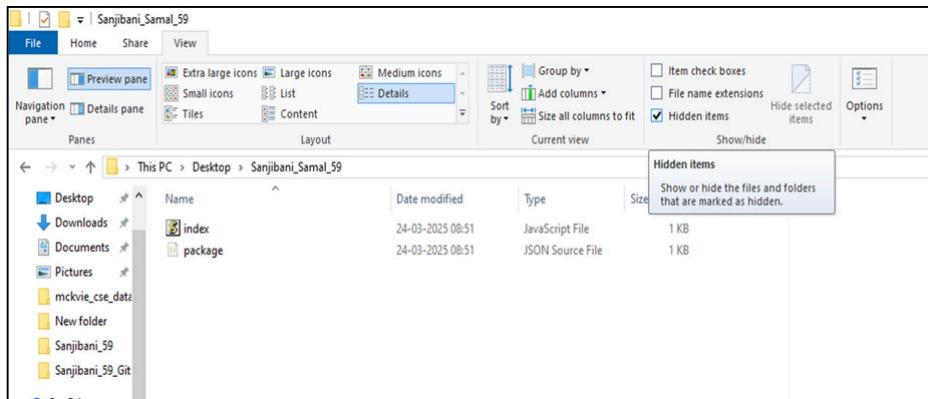


ASSIGNMENT NO :- 09

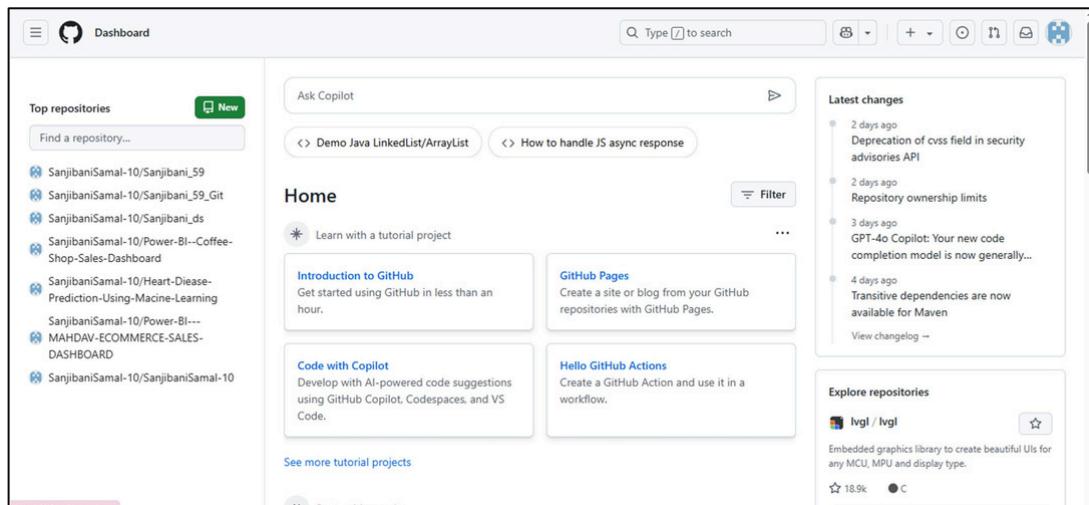
TITLE :- Deploy a project from GitHub to EC2.

STEP 01 :-

Create two files Index.js and package.json and store it in a folder here the folder here **Swastik_Ghosh_019** and then go to View Section of the folder and select Hidden Items.



STEP 02 :- Then Log in to your GitHub Account and click on New in the left handside of the page to create New Repository.



STEP 03 :- Give a name to your Repository and select Public then click on Create Repository to create the Repository.

Create a new repository
A repository contains all project files, including the revision history. Already have a project repository elsewhere?
[Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner * **Repository name ***

SanjibaniSamal-10 /

Great repository names are short and memorable. Need inspiration? How about [cautious-octo-guide](#)?

Description (optional)

Public
Anyone on the internet can see this repository. You choose who can commit.

Private
You choose who can see and commit to this repository.

Initialize this repository with:
 Add a README file
This is where you can write a long description for your project. [Learn more about READMEs](#).

Add .gitignore

Choose which files not to track from a list of templates. [Learn more about ignoring files](#).

Choose a license

A license tells others what they can and can't do with your code. [Learn more about licenses](#).

① You are creating a public repository in your personal account.

Create repository

© 2025 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Docs](#) [Contact](#) [Manage cookies](#) [Do not share my personal information](#)

STEP 04 :- A New Repository is created ,Now copy the Http Link given in the blue Box.

SanjibaniSamal-10 / Sanjibani_Samal_59

Sanjibani_Samal_59 Public

Set up GitHub Copilot
Use GitHub's AI pair programmer to autocomplete suggestions as you code.
[Get started with GitHub Copilot](#)

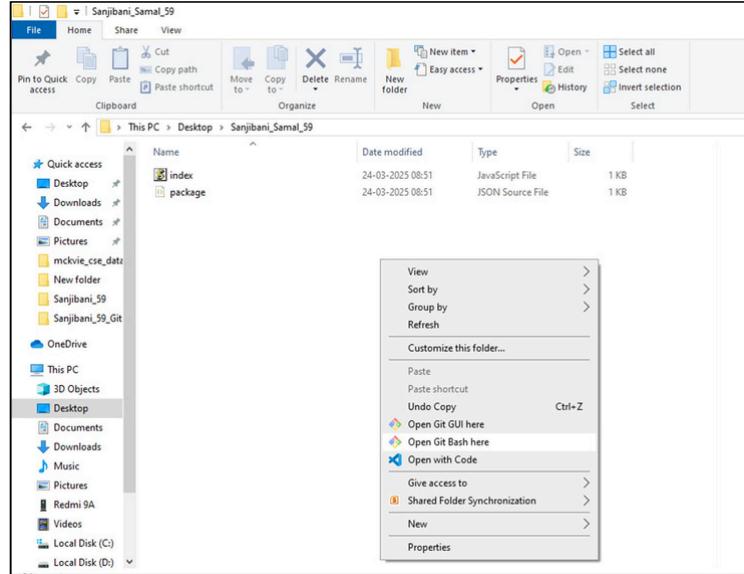
Add collaborators to this repository
Search for people using their GitHub username or email address.
[Invite collaborators](#)

Quick setup — if you've done this kind of thing before
 or https://github.com/SanjibaniSamal-10/Sanjibani_Samal_59.git
Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# Sanjibani_Samal_59" >> README.md
```

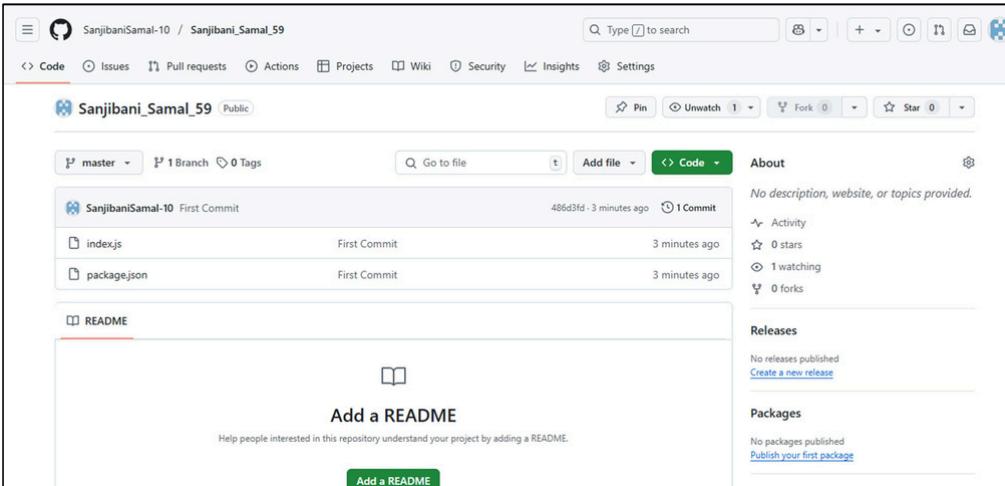
STEP 05 :- Open your folder where you have stored your files then right click on the folder and open it with Git Bash



STEP 06 :- In the Git Bash write the following command to upload your files to the GitHub Repository

```
MINGW64:/c/Users/personal/Desktop/Sanjibani_Samal_59
personal@DESKTOP-MR559VJ MINGW64 ~/Desktop/Sanjibani_Samal_59
$ git init
Initialized empty Git repository in C:/Users/personal/Desktop/Sanjibani_Samal_59/.git/
personal@DESKTOP-MR559VJ MINGW64 ~/Desktop/Sanjibani_Samal_59 (master)
$ git add .
warning: in the working copy of 'index.js', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'package.json', LF will be replaced by CRLF the next time Git touches it
personal@DESKTOP-MR559VJ MINGW64 ~/Desktop/Sanjibani_Samal_59 (master)
$ git commit -m "First Commit"
[master (root-commit) 486d3fd] First Commit
 2 files changed, 30 insertions(+)
   create mode 100644 index.js
   create mode 100644 package.json
personal@DESKTOP-MR559VJ MINGW64 ~/Desktop/Sanjibani_Samal_59 (master)
$ git remote add origin https://github.com/SanjibaniSamal-10/Sanjibani_Samal_59.git
personal@DESKTOP-MR559VJ MINGW64 ~/Desktop/Sanjibani_Samal_59 (master)
$ git push -u origin master
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 607 bytes | 151.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/SanjibaniSamal-10/Sanjibani_Samal_59.git
 * [new branch]  master -> master
branch 'master' set up to track 'origin/master'.
personal@DESKTOP-MR559VJ MINGW64 ~/Desktop/Sanjibani_Samal_59 (master)
$
```

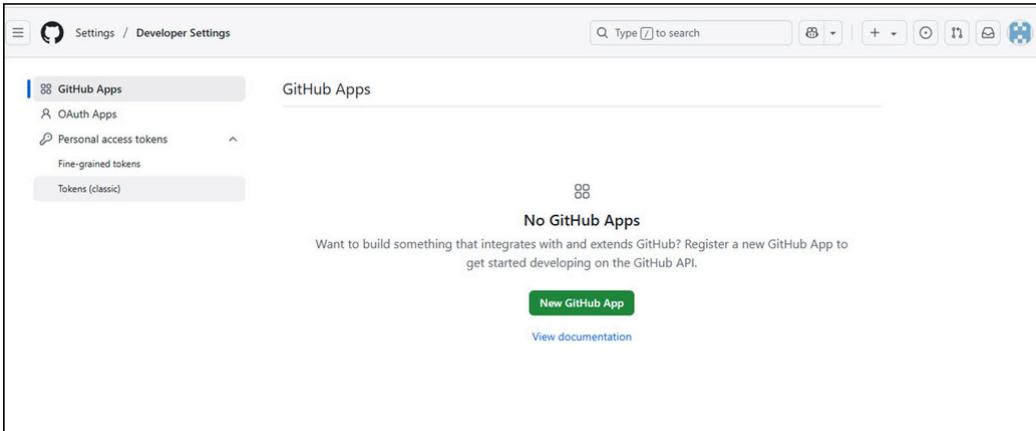
STEP 07 :- Refresh your GitHub Page and you can see your files in the Repository.



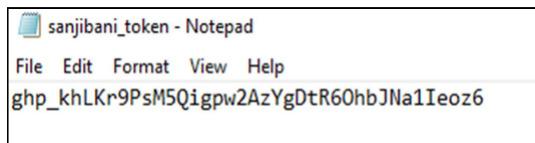
STEP 08 :- Go to your GitHub Profile and click on Settings and then goto Developer Settings

STEP 09 :- Then in Developer Setting click on Personal access tokens and then select Tokens (classic) and click on Generate Token and select all check boxes and give a

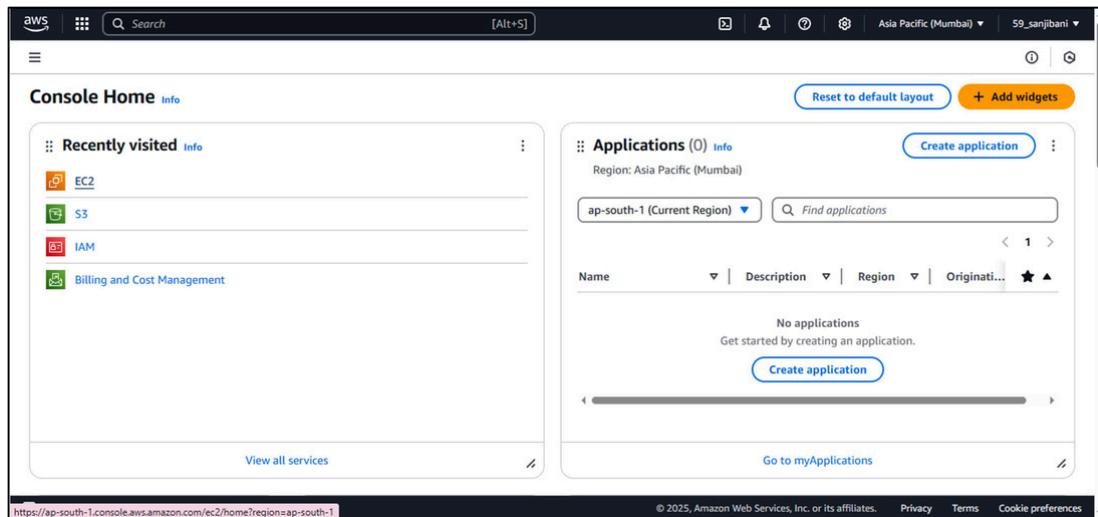
Password , Copy the generated token and save it in notepad. (Here I have already created the Token previously)



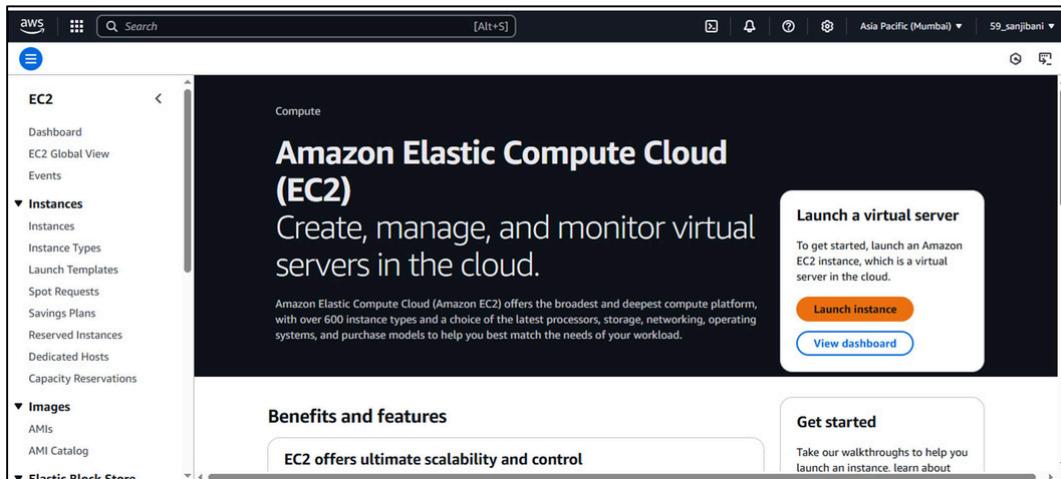
Generated Token :-



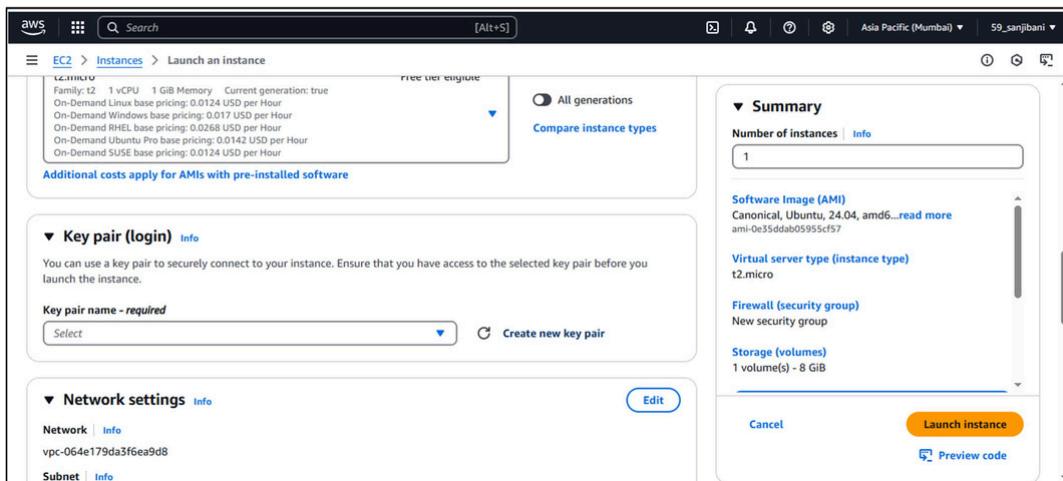
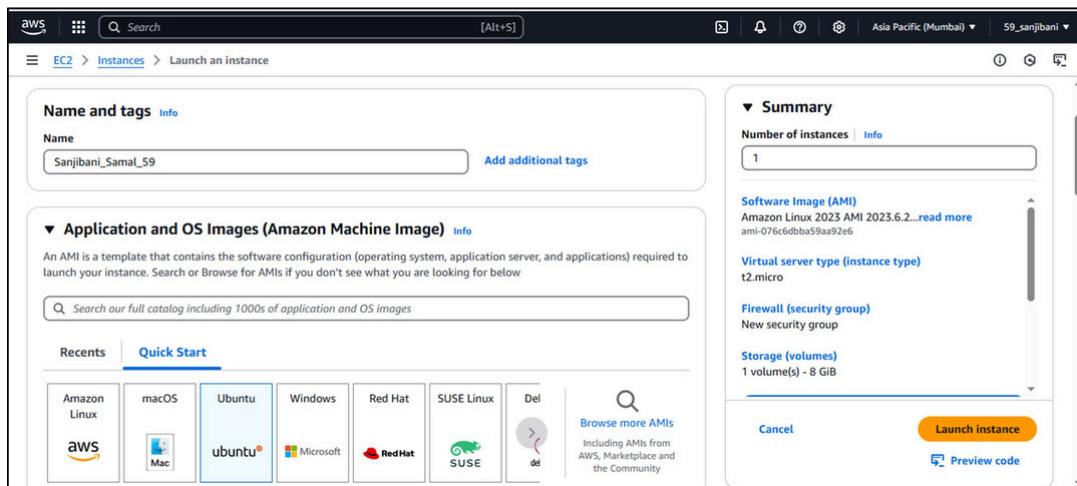
STEP 10 :- Login to your AWS Account and select EC2.

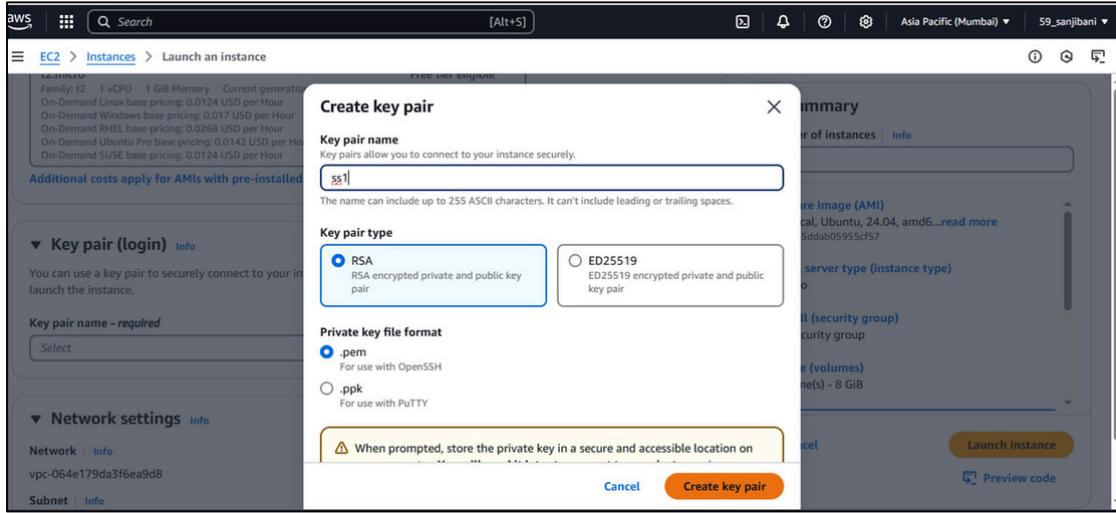


STEP 11 :- Click on Launch Instance

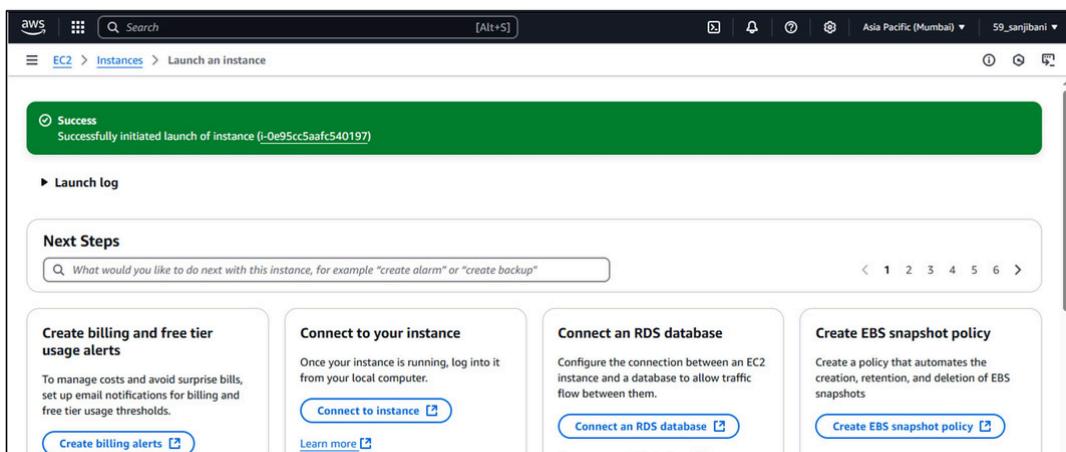
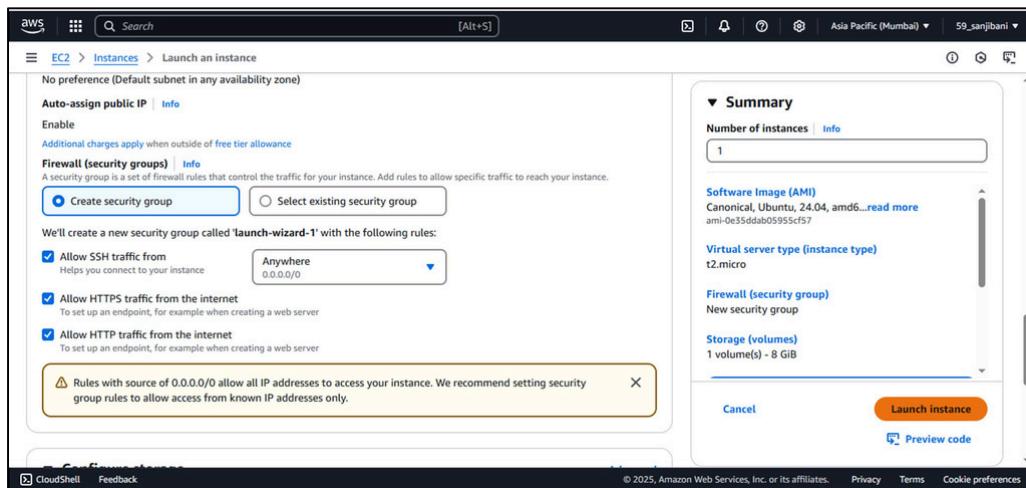


STEP 12 :- Give your instance a name and then select Ubuntu and then click on Create key pair a pop up window will appear there write the key name then select RSA and then click on .pem and click on Create Key pair at bottom of page.





STEP 13 :- Select Create security group and select all the checkboxes of SSH , HTTPPs and HTTP then click on Launch Instance at the right hand side bottom of page and you have successfully created your Instance.



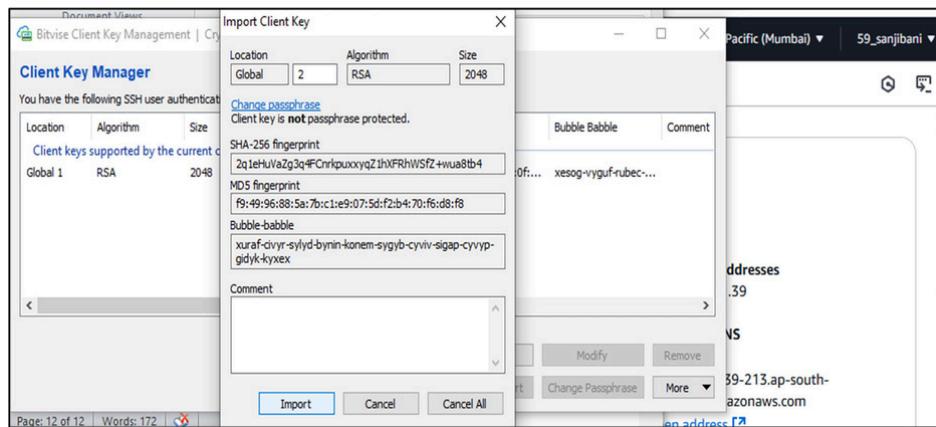
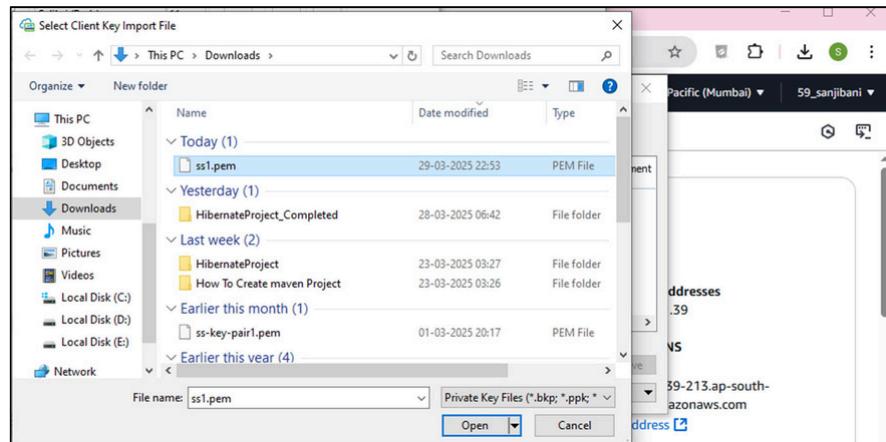
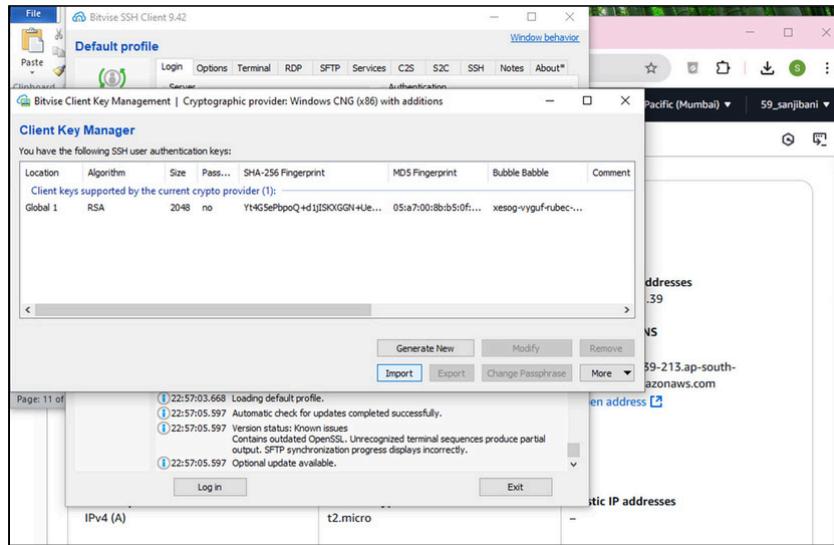
STEP 14 :- Click on the Instance ID and copy the Public IPv4 Address.

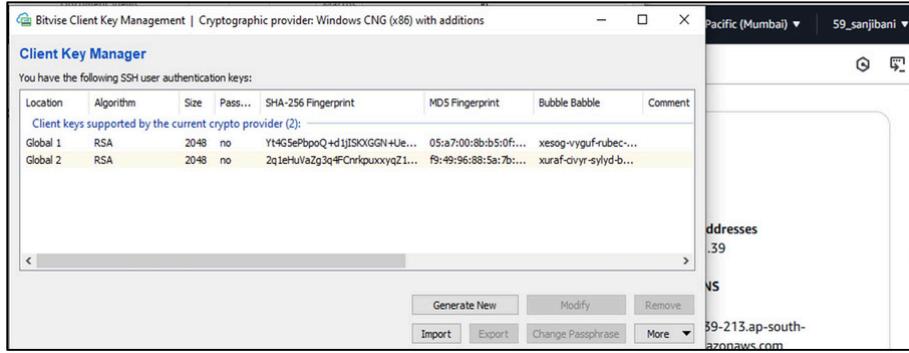
The screenshot shows the AWS EC2 Instances page. A single instance named "Sanjibani_Sam..." is listed with the instance ID "i-0e95cc5aafc540197". The instance is running and is a t2.micro type. The Public IPv4 address, "13.126.239.213", is highlighted in green with a tooltip indicating it has been copied. The left sidebar shows various EC2 management options like Dashboard, Instance Types, and Launch Templates.

STEP 15 :- Open the Bitvise SSH Client Server and paste the Public IPv4 Address in the Host and then click on Client Key Manager

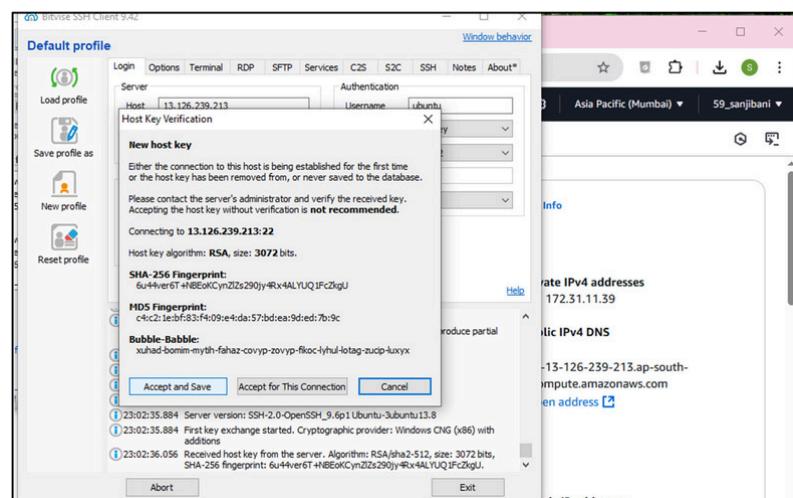
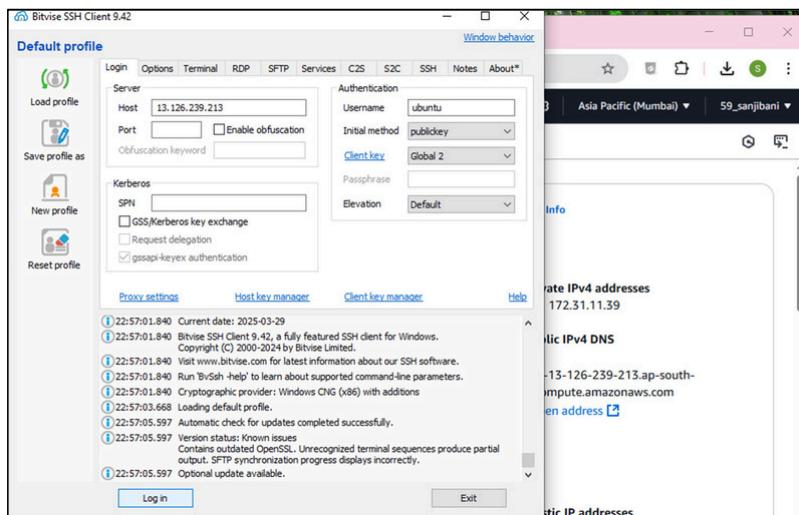
The screenshot shows the Bitvise SSH Client interface. The host field is set to "13.126.239.213". The "Client key manager" tab is selected in the bottom navigation bar. The status bar at the bottom right shows "22:57:01.840 Current date: 2025-02-29". The right pane displays the AWS EC2 instance details, including the Public IPv4 address "13.126.239.213.ap-south-1.compute.amazonaws.com" and the Private IP DNS name "ip-172-31-11-39.ap-south-1.compute.internal".

STEP 15 :- In Client Key Manager click on Import then select your Downloaded key file and click on Import your key is named as Global Key (here) then again click on import

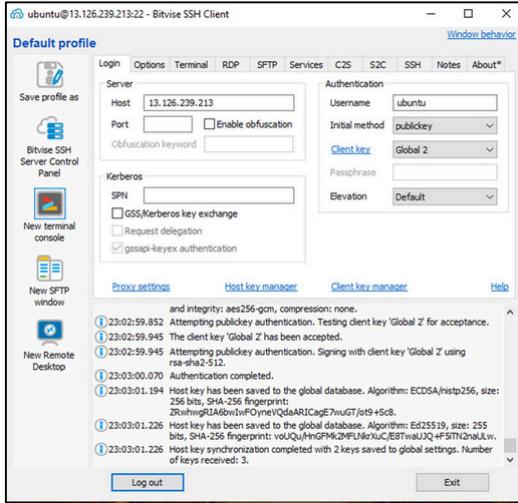




STEP 16 :- On the Right handside Give the Username as Ubuntu ,Initial method as publickey and clientkey as Global2 and then click on Log In and after that click on Accept key and Save



STEP 17 :- Then click on Log In Terminal of Bitvise SSH Client Server



STEP 18 :- Write the following Command in the terminal and run it .

- sudo apt-get update
- sudo apt-get upgrade
- sudo apt-get install nginx
- curl
- sudo apt install nodejs
- node -v

```
ubuntu@ip-172-31-11-39:~$ sudo apt-get update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:5 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
```

```
ubuntu@ip-172-31-11-39:~$ sudo apt-get upgrade
Get:41 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [711 kB]
Get:42 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [136 kB]
Get:43 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [8992 B]
Get:44 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [7068 B]
Get:45 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [822 kB]
Get:46 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [177 kB]
Get:47 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [52.0 kB]
Get:48 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [17.0 kB]
Get:49 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [810 kB]
Get:50 http://security.ubuntu.com/ubuntu noble-security/restricted Translation-en [164 kB]
Get:51 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [212 B]
Get:52 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 c-n-f Metadata [468 B]
Get:53 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [17.6 kB]
Get:54 http://security.ubuntu.com/ubuntu noble-security/multiverse Translation-en [3792 B]
Get:55 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [208 B]
Get:56 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [380 B]
Fetched 33.1 MB in 23s (1419 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages have been kept back:
  linux-headers-4.15.0-102-generic
```

```
ubuntu@13.126.239.213:22 - Bitvise xterm - ubuntu@ip-172-31-11-39:~ 
systemctl restart unattended-upgrades.service
No containers need to be restarted.

User sessions running outdated binaries:
ubuntu @ session #2: sshd[1088]
ubuntu @ user manager service: systemd[1093]
ubuntu@ip-172-31-11-39:~$ 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
```

```
ubuntu@3.108.217.127:22 - Bitvise xterm - ubuntu@ip-172-31-9-4:~ 
Service restarts being deferred:
/etc/needrestart/restart.d/dbus.service
systemctl restart networkd-dispatcher.service
systemctl restart systemd-logind.service
systemctl restart unattended-upgrades.service

No containers need to be restarted.

User sessions running outdated binaries:
ubuntu @ session #1: sshd[1026]
ubuntu @ user manager service: systemd[1034]

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-9-4:~$ curl -sL https://deb.nodesource.com/setup_18.x | sudo -E bash -
2025-03-30 06:30:15 - Installing pre-requisites
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

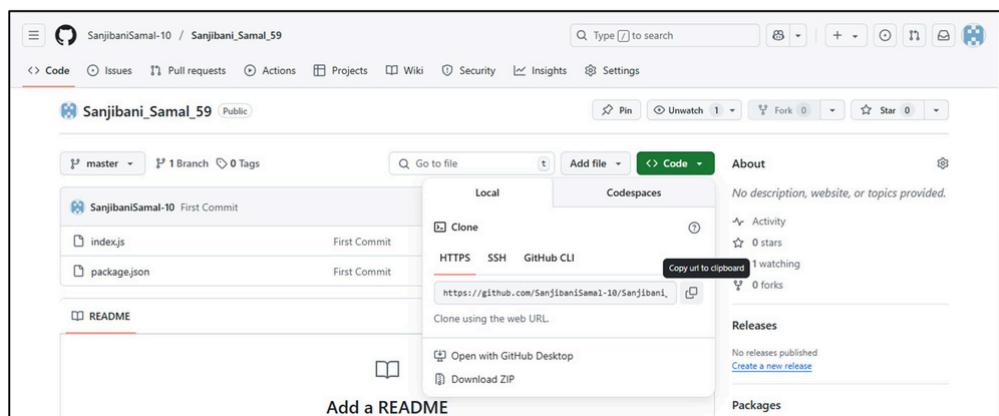
```
ubuntu@3.108.217.127:22 - Bitvise xterm - ubuntu@ip-172-31-9-4:~/Sanjibani_Samal_59
2025-03-30 06:30:21 - You can use N|solid Runtime as a node.js alternative
2025-03-30 06:30:21 - To install N|solid Runtime, run: apt-get install nsolid -y

ubuntu@ip-172-31-9-4:~$ sudo apt install nodejs
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  nodejs
0 upgraded, 1 newly installed, 0 to remove and 6 not upgraded.
Need to get 29.7 MB of archives.
After this operation, 187 MB of additional disk space will be used.
Get:1 https://deb.nodesource.com/node_18.x nodistro/main amd64 nodejs amd64 18.20.8-1no
.7 MB]
```

```
User sessions running outdated binaries:
ubuntu @ session #1: sshd[1026]
ubuntu @ user manager service: systemd[1034]

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-9-4:~$ node -v
v18.20.8
```

STEP 19 :- Goto to your GitHub Account Copy you Repository URL by clicking on Code



STEP 20 :- Then write the command in terminal and run it

- git clone <Repository link>
- cd <repository name>
- ls
- npm install

```
ubuntu@ip-172-31-9-4:~$ git clone https://github.com/SanjibaniSamal-10/Sanjibani_Samal_59
Cloning into 'Sanjibani_Samal_59'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 4 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (4/4), done.
```

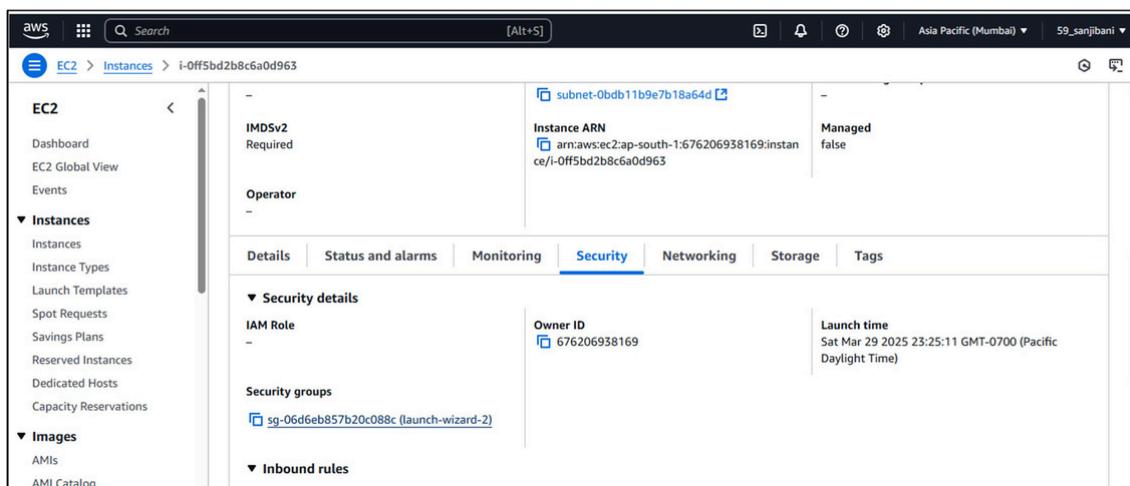
```
ubuntu@ip-172-31-9-4:~$ cd Sanjibani_Samal_59
ubuntu@ip-172-31-9-4:~/Sanjibani_Samal_59$ ls
index.js package.json
```

```
index.js package.json
ubuntu@ip-172-31-9-4:~/Sanjibani_Samal_59$ npm install
added 227 packages, and audited 228 packages in 14s

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

found 0 vulnerabilities
npm notice
npm notice New major version of npm available! 10.8.2 -> 11.2.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v11.2.0
npm notice To update run: npm install -g npm@11.2.0
npm notice
  Audit report generated by running: npm audit --production --force
```

STEP 21 :- Then Go to your Instance ID and click on Security and then click on Security groups link



STEP 22 :- Then click on Edit Inbound Rule

The screenshot shows the AWS EC2 Security Groups page. The security group selected is "sg-06d6eb857b20c088c - launch-wizard-2". The "Details" section displays the security group name, ID, description, owner, and rule counts. The "Inbound rules" tab is selected, showing three existing rules: one for SSH (TCP port 22) from 0.0.0.0/0, one for HTTPS (TCP port 443) from 0.0.0.0/0, and one for HTTP (TCP port 80) from 0.0.0.0/0.

STEP 23:- Then click on Add rule

The screenshot shows the "Edit inbound rules" page for the "sg-06d6eb857b20c088c - launch-wizard-2" security group. A new rule is being added with the following configuration:

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-0072d605eb91b25a5	SSH	TCP	22	Cus... (0.0.0.0/0)	Allow SSH access
sgr-02d43820021af8c47	HTTPS	TCP	443	Cus... (0.0.0.0/0)	Allow HTTPS access
sgr-0cd8cdf48a7454340	HTTP	TCP	80	Cus... (0.0.0.0/0)	Allow HTTP access

An "Add rule" button is visible at the bottom left. A warning message at the bottom states: "⚠️ Rules with source of 0.0.0.0/0 or -/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only."

>> Give the Port Range as 4000 and 0.0.0.0/0

Inbound rules [Info](#)

Security group rule ID	Type Info	Protocol Info	Port range	Source Info	Description - optional Info
sgr-0072d605eb91b25a5	SSH	TCP	22	Cus... Info	<input type="text"/> 0.0.0.0/0 X
sgr-02d43820021af8c47	HTTPS	TCP	443	Cus... Info	<input type="text"/> 0.0.0.0/0 X
sgr-0cd8cdf48a7454340	HTTP	TCP	80	Cus... Info	<input type="text"/> 0.0.0.0/0 X
-	Custom TCP	TCP	4000	Any... Info	<input type="text"/> 0.0.0.0/0 X

[Add rule](#)

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

STEP 24 :- Then click on Save Changes

Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

[Cancel](#) [Preview changes](#) [Save rules](#)

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

>> Your security group is changed Successfully

The screenshot shows the AWS EC2 Security Groups interface. A green success message at the top states: "Inbound security group rules successfully modified on security group (sg-06d6eb857b20c088c | launch-wizard-2) Details". Below this, the security group details are listed: Security group name: launch-wizard-2, Security group ID: sg-06d6eb857b20c088c, Description: launch-wizard-2 created 2025-03-30T06:24:30.534Z, Owner: 676206938169, Inbound rules count: 4 Permission entries, Outbound rules count: 1 Permission entry. The "Inbound rules" tab is selected. At the bottom, there are tabs for Outbound rules, Sharing - new, VPC associations - new, and Tags. The footer includes copyright information and links for Privacy, Terms, and Cookie preferences.

STEP 25 :- Write node index.js in terminal to Start the Server

```
round ✓ vulnerabilities
npm notice
npm notice New major version of npm available! 10.8.2 -> 11.2.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v11.2.0
npm notice To update run: npm install -g npm@11.2.0
npm notice
ubuntu@ip-172-31-9-4:~/Sanjibani_Samal_59$ node index.js
Started server
```

STEP 26 :- Then copy the Instance Public IPV4 Address and paste in Incognito Window concating the IPV4 Address with : 4000 (port range)

The screenshot shows the AWS EC2 Instances page. It displays an instance summary for "i-Off5bd2b8c6a0d963 (Sanjibani_Samal_59)". The instance state is "Running". The public IPv4 address is listed as "3.108.217.127 | open address". A context menu is open over this address, with the option "Public IPv4 address copied" highlighted in green. The browser's right-click context menu is also visible, showing options like "New tab", "New window", "New Incognito window", and "Private browsing". Other items in the menu include "Passwords and autofill", "History", "Downloads", "Bookmarks and lists", "Tab groups", "Extensions", "Delete browsing data...", "Zoom", "Print...", "Search with Google Lens", "Translate...", "Find and edit", and "Cast, save, and share".

