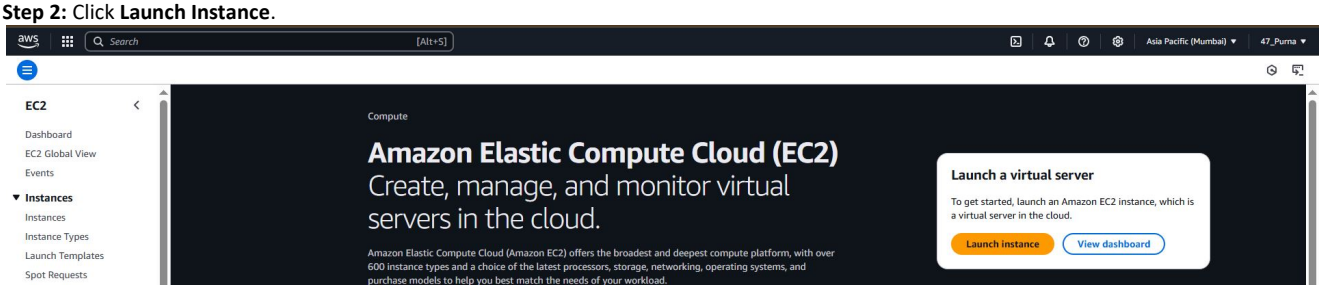
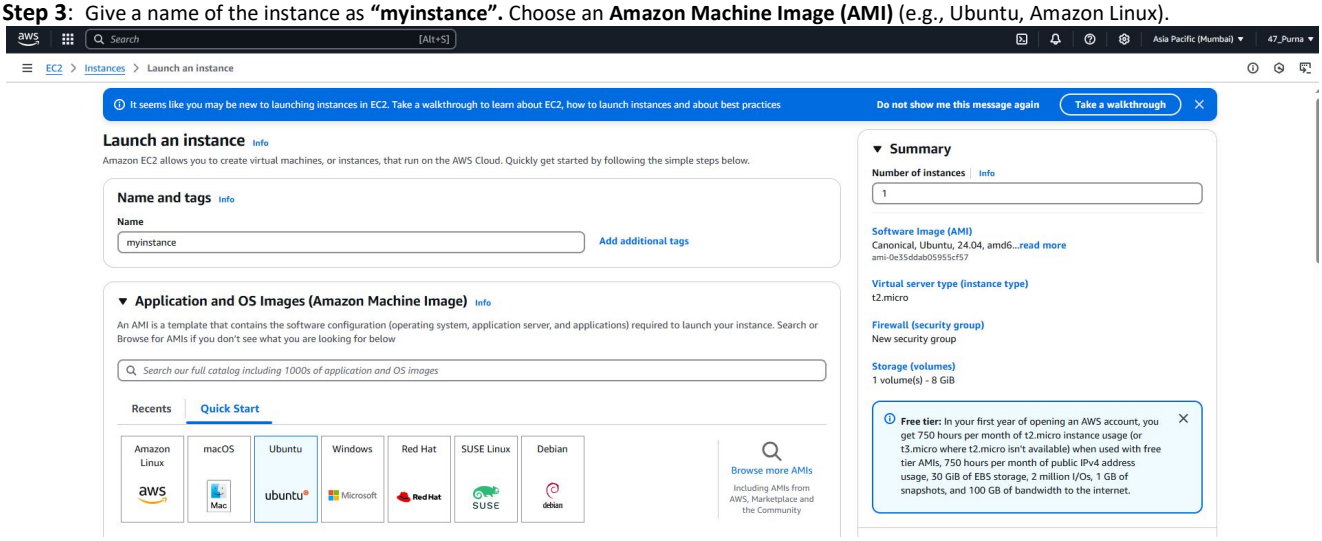
## Assignment 9: Deploy a project from Github to Ec2.

**Step 1 : Log in** to AWS account .To Launch an EC2 Instance Navigate to **EC2 Dashboard**.


**Step 2:** Click **Launch Instance**.



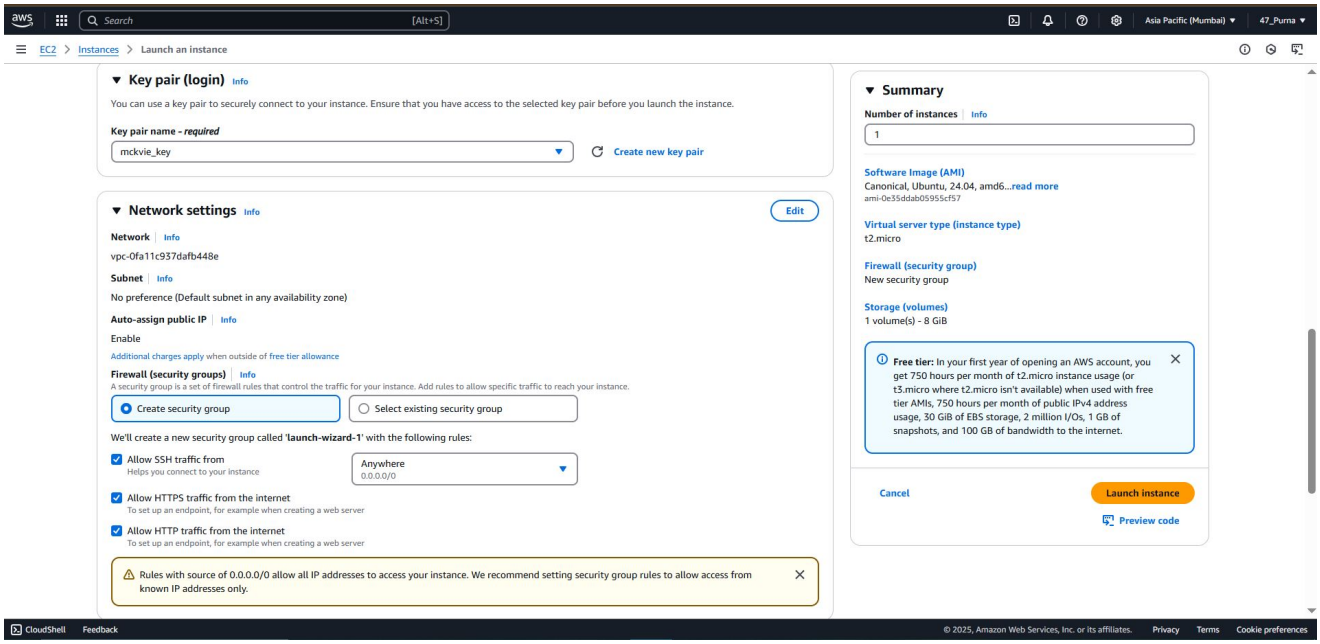

**Step 3**: Give a name of the instance as **"myinstance".** Choose an **Amazon Machine Image (AMI)** (e.g., Ubuntu, Amazon Linux).



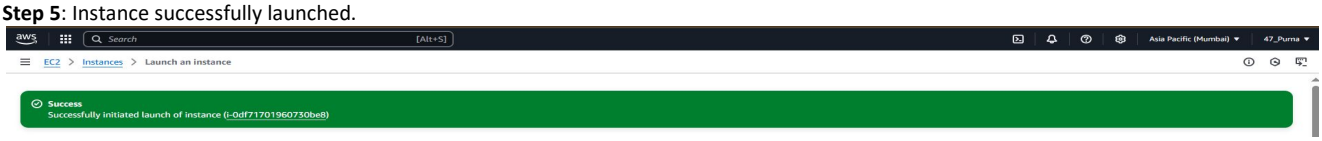

**Step 4:** Choose a key pair or create a key pair. To Configure Security Group:

- Allow SSH (Port 22) from IP.
- To hosting a web app, allow HTTP (Port 80) or HTTPS (Port 443).

And click on "**Launch instance**".

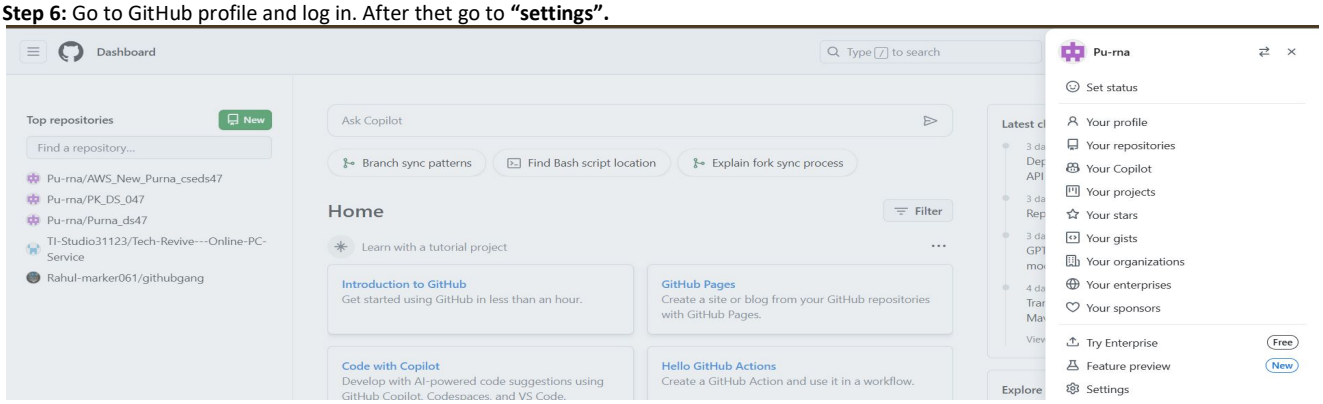



**Step 5**: Instance successfully launched.

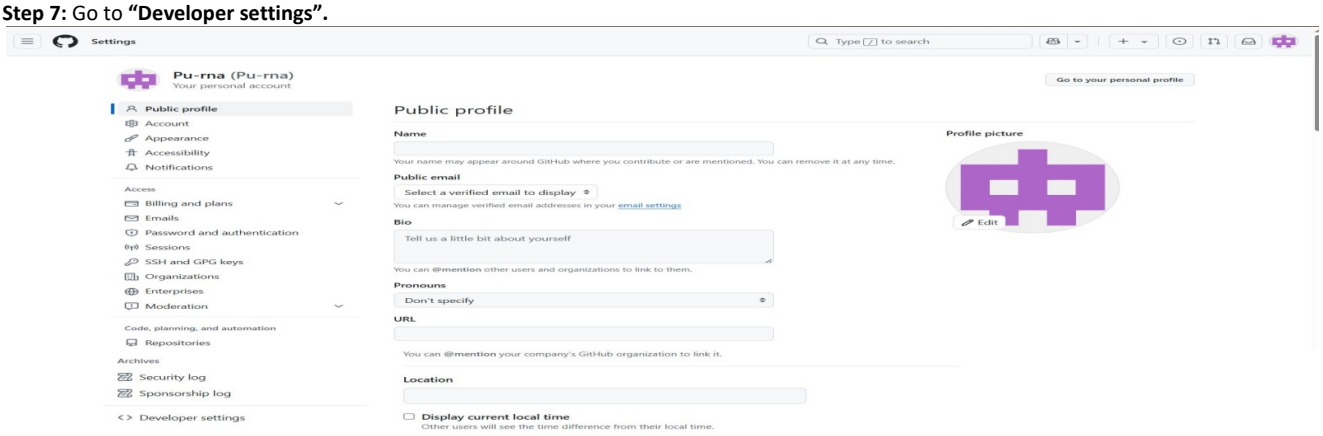



Purna Kundu/DS/22/047

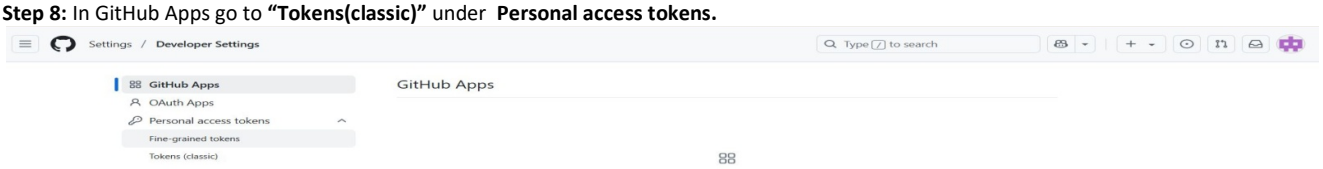
**Step 6:** Go to GitHub profile and log in. After thet go to **“settings”.**




**Step 7:** Go to **“Developer settings”.**

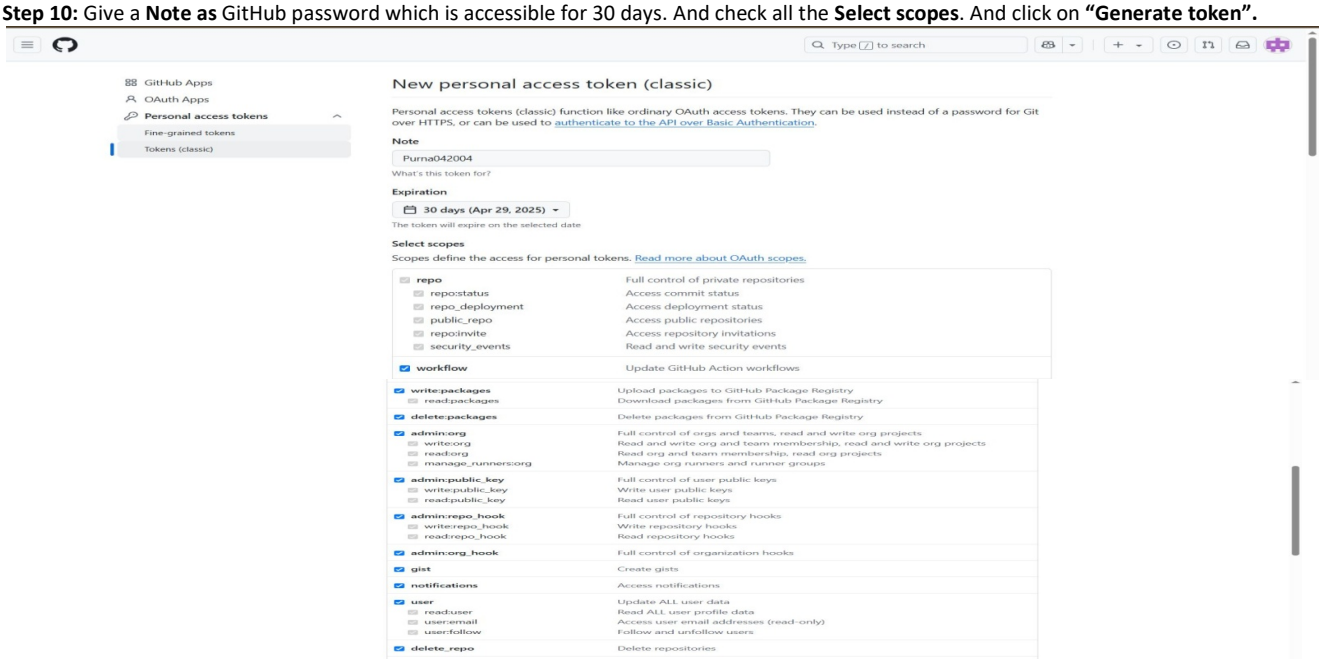



**Step 8:** In GitHub Apps go to **“Tokens(classic)”** under **Personal access tokens.**

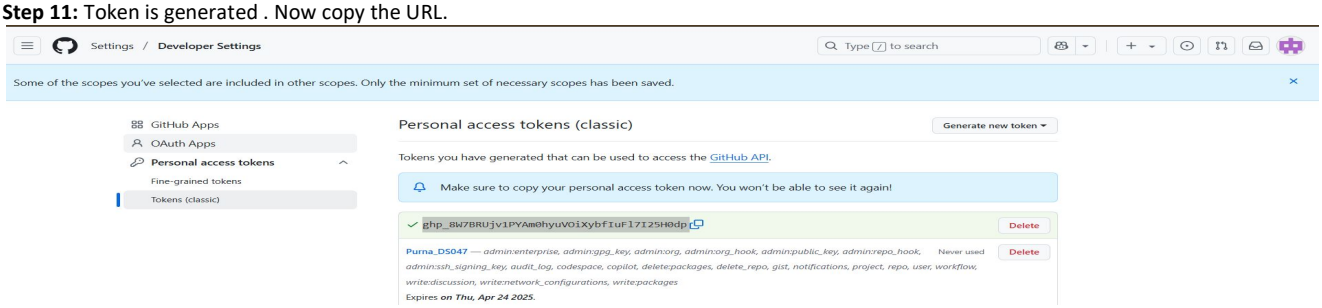



**Step 9**: Click on **“Generate new token(classic)”** under **Generate new token.**

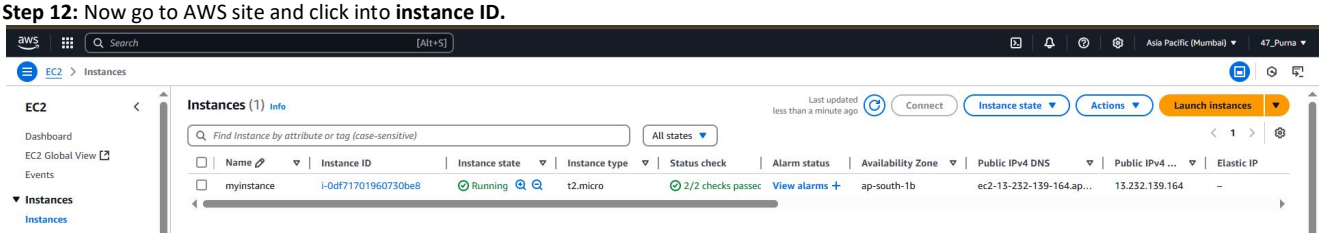

**Step 10:** Give a **Note as** GitHub password which is accessible for 30 days. And check all the **Select scopes**. And click on **“Generate token”.**



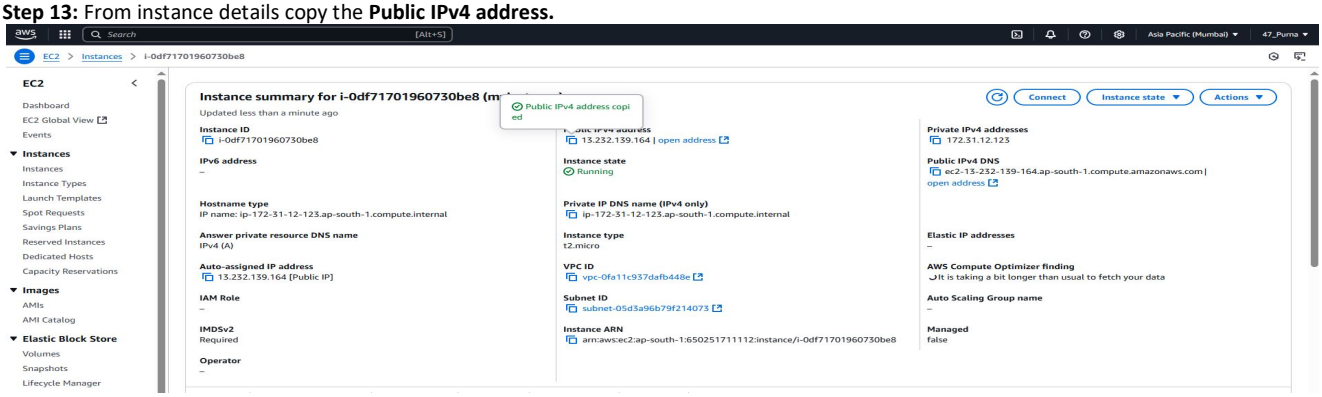

**Step 11:** Token is generated . Now copy the URL.



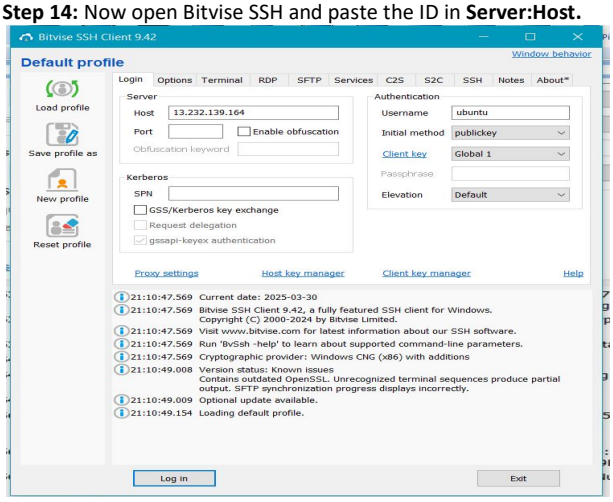

**Step 12:** Now go to AWS site and click into **instance ID.**



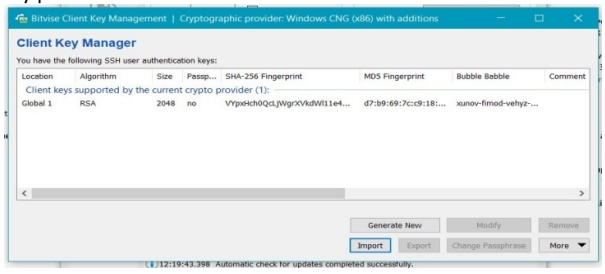

**Step 13:** From instance details copy the **Public IPv4 address.**



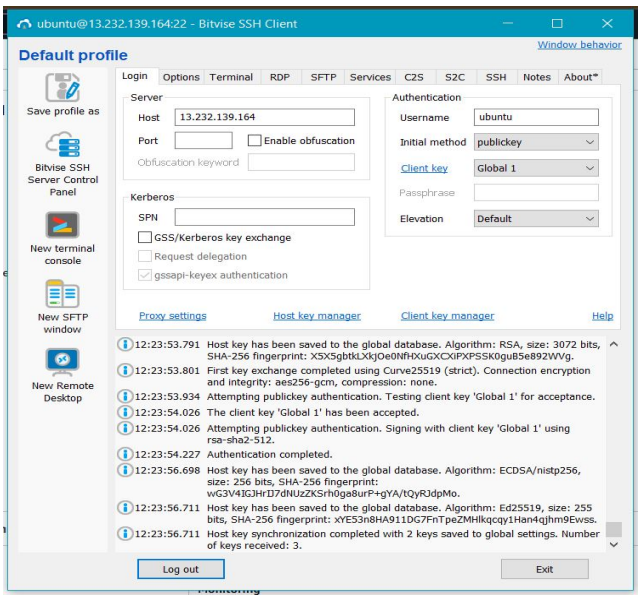

**Step 14:** Now open Bitvise SSH and paste the ID in **Server:Host.**




**Step 15:** Now click on **Client key manager** and import key pair which is use to launched instance. After tha close the tab.




**Step 16:** Now click on log in. And open **New terminal console.**




Purna Kundu/DS/22/047

**Step 17:** Once connected to EC2 instance, install necessary software or Dependencies. For that give this necessary command step by step:

- 
```
ubuntu@ip-172-31-12-123:~$ sudo apt-get install update
```

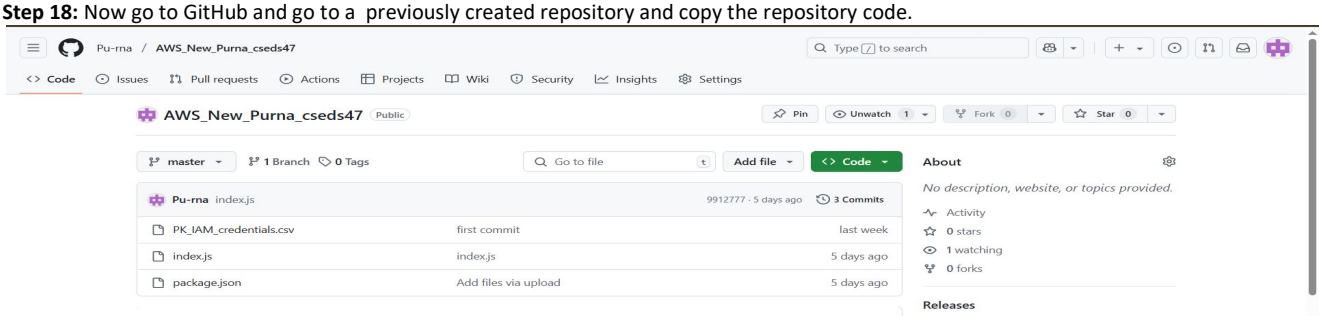
- 
```
ubuntu@ip-172-31-12-123:~$ sudo apt-get install upgrade
```

- 
```
ubuntu@ip-172-31-12-123:~$ sudo apt-get install nginx
ubuntu@ip-172-31-12-123:~$ nginx -v
nginx version: nginx/1.24.0 (Ubuntu)
ubuntu@ip-172-31-12-123:~$
```

- 
```
ubuntu@ip-172-31-12-123:~$ curl -sL https://deb.nodesource.com/setup_18.x | sudo -E bash -
2025-03-30 07:06:18 - Installing pre-requisites
Reading package lists... Done
2025-03-30 07:06:54 - Repository configured successfully.
2025-03-30 07:06:54 - To install Node.js, run: apt-get install nodejs -y
2025-03-30 07:06:54 - You can use N|solid Runtime as a node.js alternative
2025-03-30 07:06:54 - To install N|solid Runtime, run: apt-get install nsolid -y
```

- 
```
ubuntu@ip-172-31-12-123:~$ sudo apt install nodejs
ubuntu@ip-172-31-12-123:~$ node -v
v18.20.8
```

**Step 18:** Now go to GitHub and go to a previously created repository and copy the repository code.



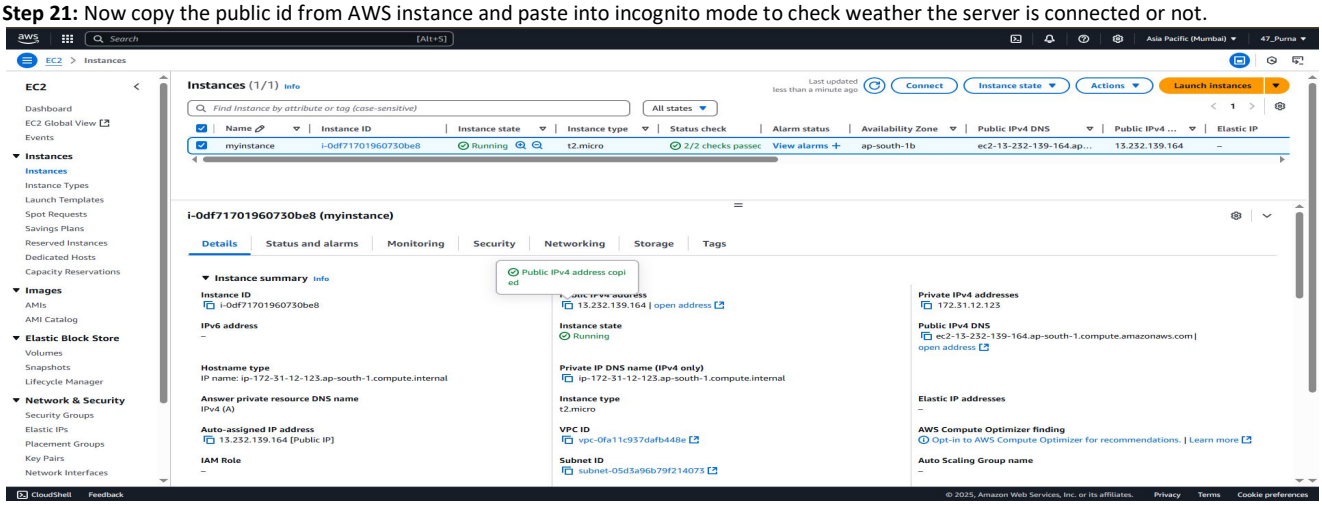

**Step 19:**And now again go to the terminal and paste the url with this command to link with github and Ec2.

```
ubuntu@ip-172-31-12-123:~$ git clone https://github.com/Pu-rna/AWS_New_Purna_cseds47.git
Cloning into 'AWS_New_Purna_cseds47'...
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 2 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (10/10), done.
Resolving deltas: 100% (2/2), done.
```

**Step 20:** Go to desired directory and install npm after that to connect with server pass command "**node index.js**" and server is started.

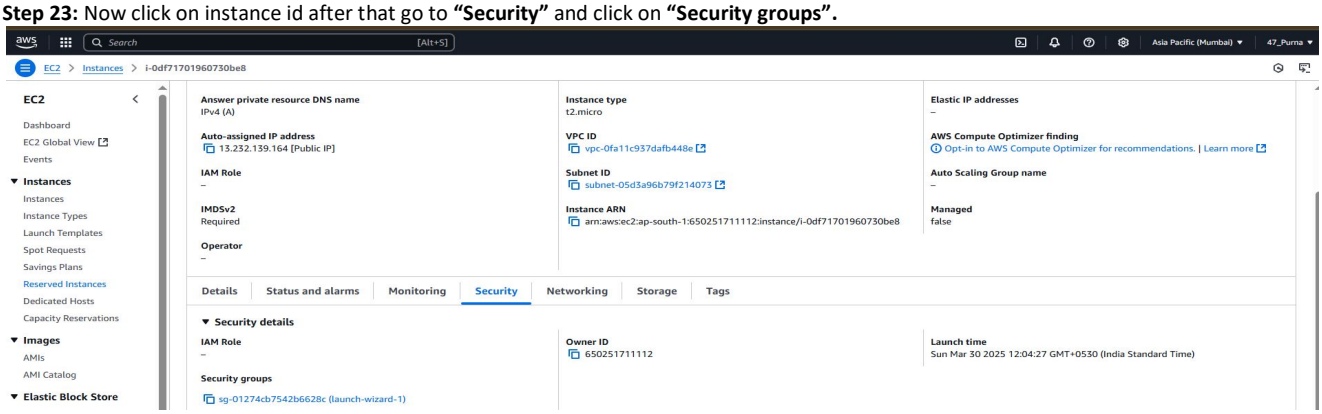
```
ubuntu@ip-172-31-12-123:~$ ls
AWS_New_Purna_cseds47
ubuntu@ip-172-31-12-123:~$ cd AWS_New_Purna_cseds47
ubuntu@ip-172-31-12-123:~/AWS_New_Purna_cseds47$ ls
PK_IAM_credentials.csv  index.js  package.json
ubuntu@ip-172-31-12-123:~/AWS_New_Purna_cseds47$ npm install

added 227 packages, and audited 228 packages in 11s

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

found 0 vulnerabilities
ubuntu@ip-172-31-12-123:~/AWS_New_Purna_cseds47$ node index.js
Started server
```

**Step 21:** Now copy the public id from AWS instance and paste into incognito mode to check weather the server is connected or not.

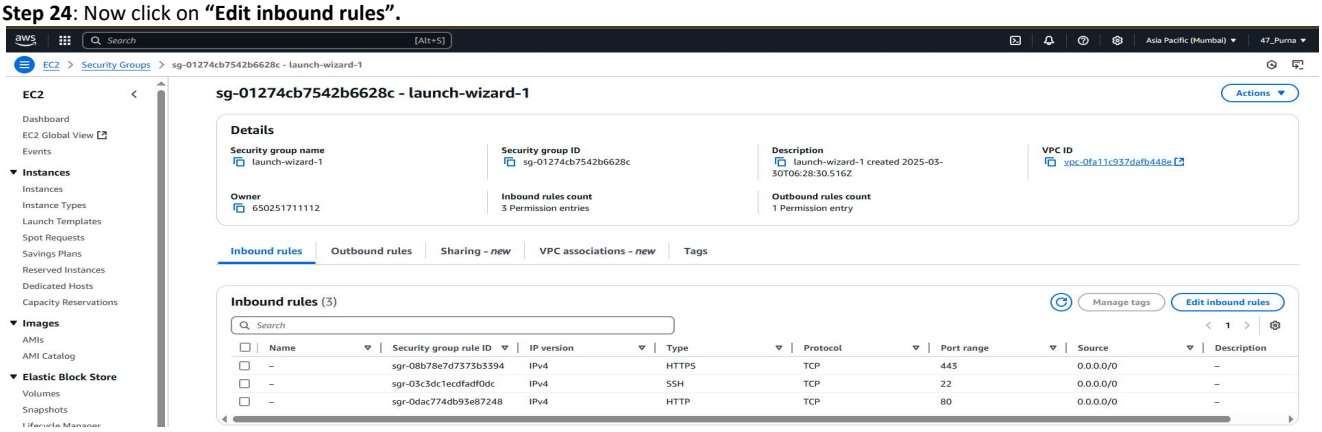



**Step 22:** Server is connected.

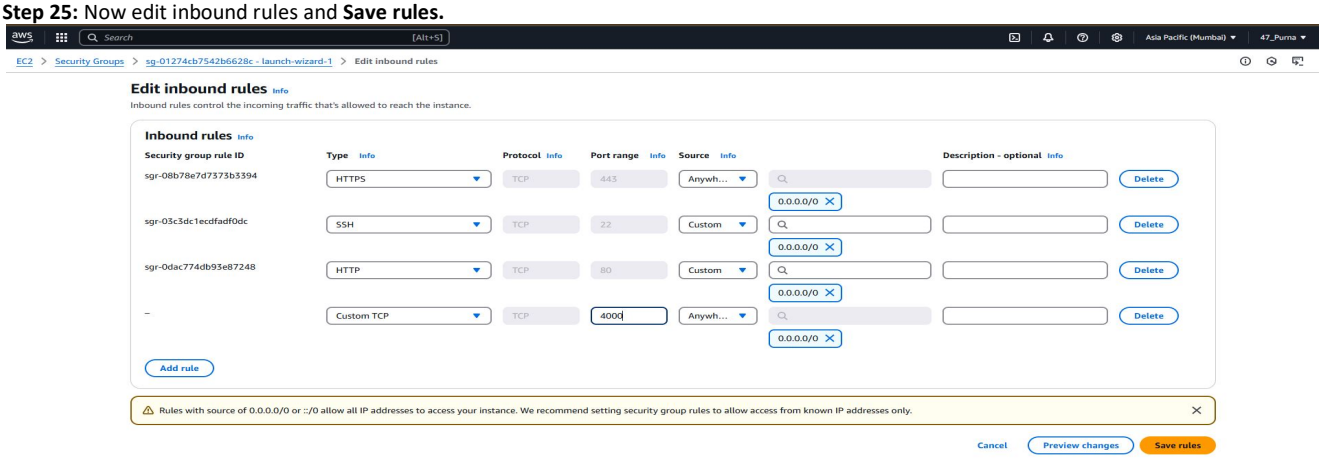

**Step 23:** Now click on instance id after that go to **"Security"** and click on **"Security groups".**



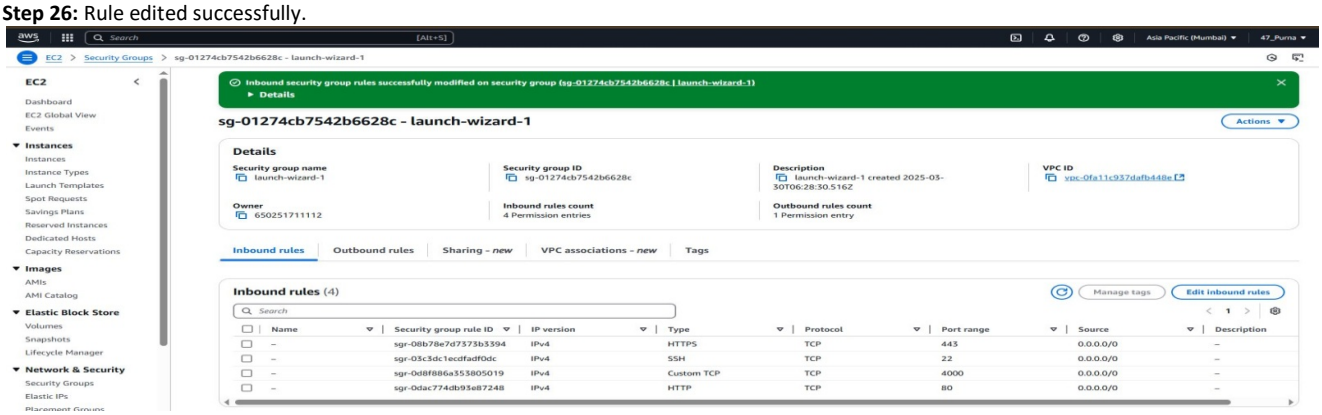

**Step 24**: Now click on **"Edit inbound rules".**




**Step 25:** Now edit inbound rules and **Save rules.**

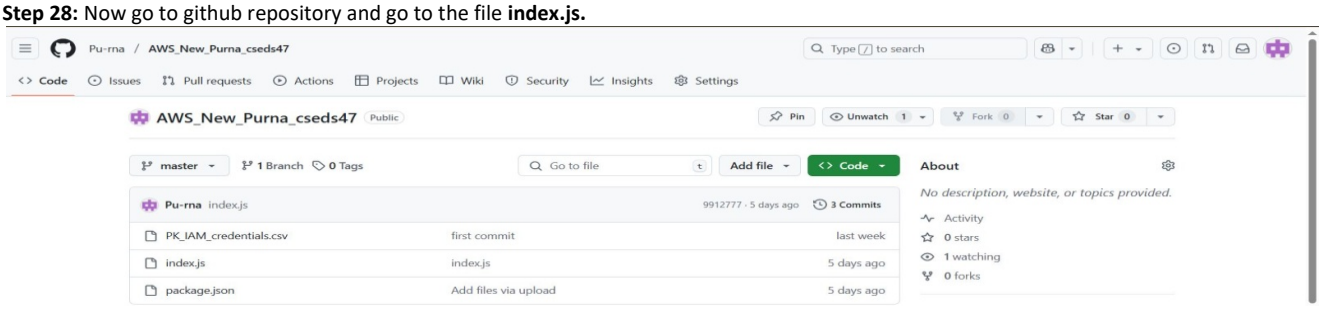



**Step 26:** Rule edited successfully.

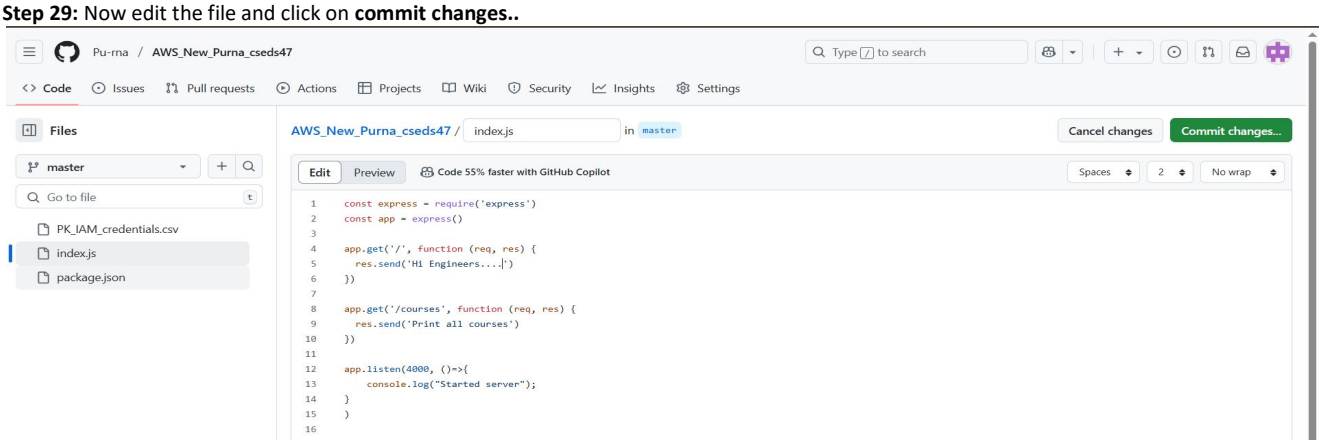



**Step 27:Now** again refresh the previous incognito tab and now the repository file is successfully accessible.

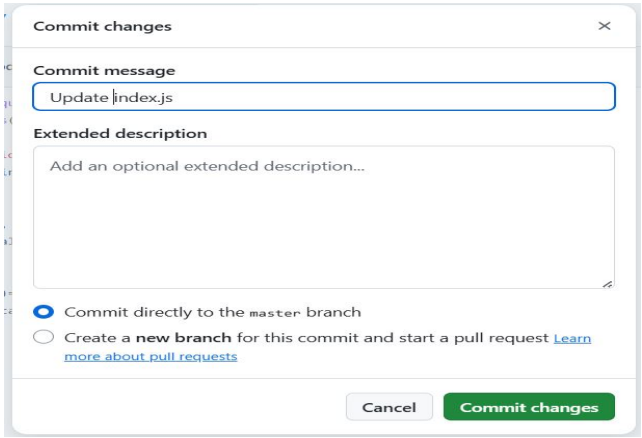

**Step 28:** Now go to github repository and go to the file **index.js.**



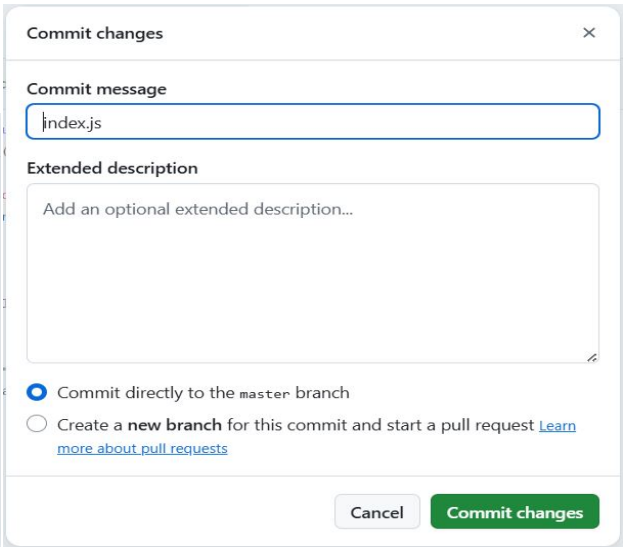

**Step 29:** Now edit the file and click on **commit changes..**




**Step 30:** Now this page is open delete **Update** from **commit message** .




**Step 31:** And click on **commit changes.**




**Step 32:** Now refresh the incognito mode and file is successfully edited.