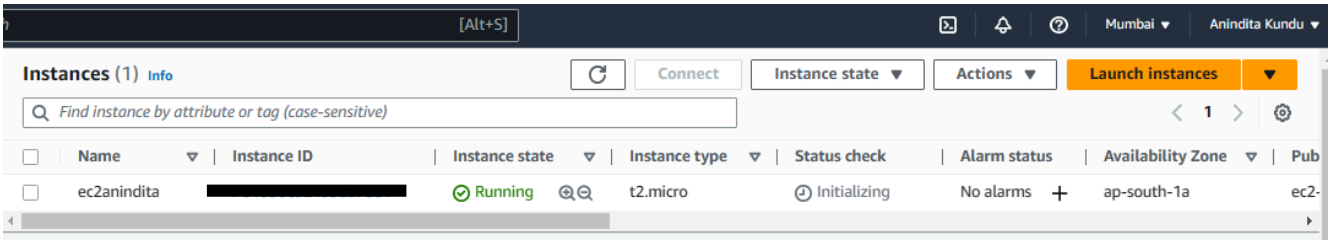
ASSIGNMENT – 9

Problem Statement: Deploy a project from GitHub to EC2.

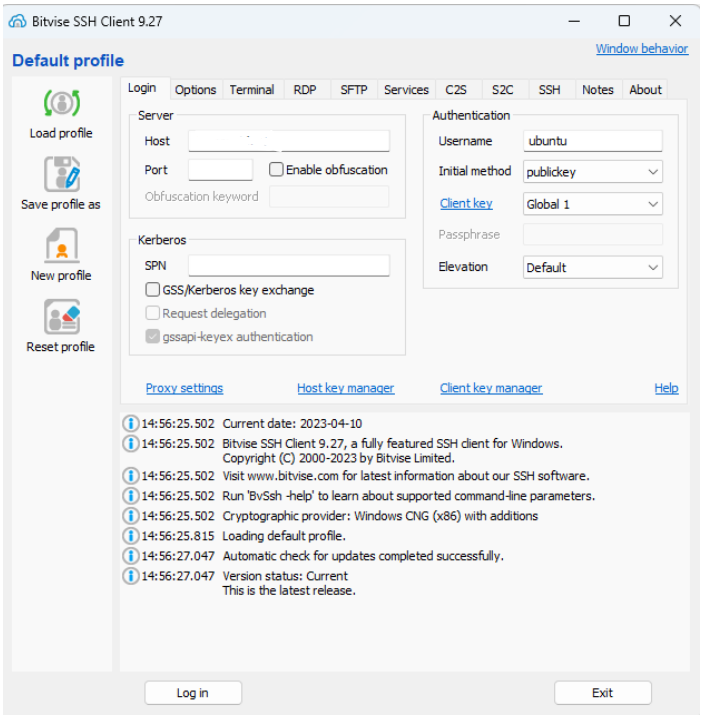
Procedure:

Step 1: Go to GitHub Website and Sign In to your account. Also, Sign-In to your AWS account.

Step 2: Create an EC2 instance (Refer to Ass7) .



Step 3: Connect the to the instance using the Bitvise SSH Client.



Step 4: Now Click on New Terminal Console option in the Left Sidebar of the Bitvise Client. A terminal window will open and in it type the following commands:-

- **sudo apt-get update**
- **sudo apt-get upgrade**

press Y and then Enter when prompted.
After the process is completed a UI appears on the screen, just press ‘Enter’ to continue.

- **sudo apt-get install nginx**

press Y and then Enter when prompted.
After the process is completed a new box/window appears. But just press Enter to continue.

- **nginx -v**

```
ubuntu@ip-172-31-34-76:~$ nginx -v
nginx version: nginx/1.18.0 (Ubuntu)
```

This command displays the nginx version installed in the server system.

- **curl -Sl https://deb.nodesource.com/setup_18.x | sudo -E bash -**

This command downloads NodeJS files with all dependencies in our server system.

- **sudo apt install nodejs**

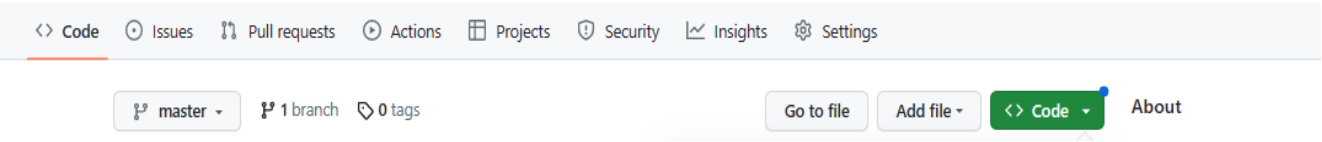
Press ‘Enter’ to continue when any UI appears on screen. This command installs NodeJS in our server system.

- **node -v**

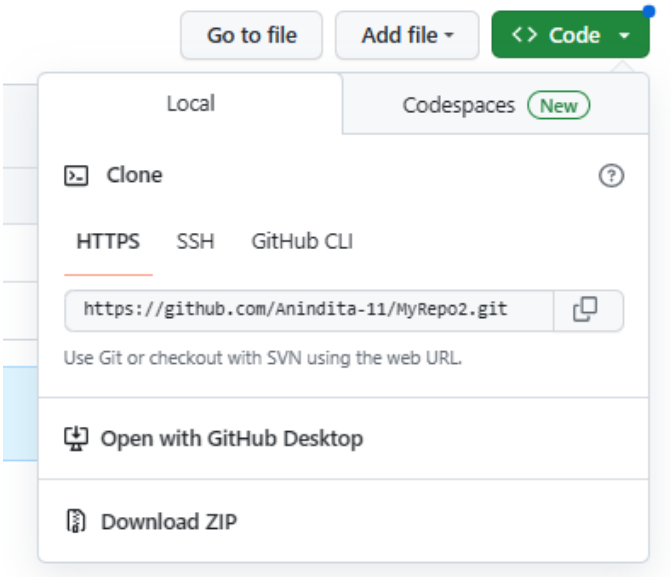
```
ubuntu@ip-172-31-34-76:~$ node -v
v18.15.0
```

This command displays the version of NodeJS installed in our server system.
Now go to the browser where your GitHub is logged in.

Step 5: Go to your GitHub Repository which you want to upload in your EC2 server. Click on the code button.



Step 6: Now copy the HTTPS address of your Repository.



Step 7: Now again go to the terminal and enter the commands:

- **git clone https-address-you-just-copied**

```
ubuntu@ip-172-31-34-76:~$ git clone https://github.com/Anindita-11/MyRepo2.git
Cloning into 'MyRepo2'...
Username for 'https://github.com': Anindita-11
Password for 'https://Anindita-11@github.com':
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 5 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.
```

After this step you will be asked to enter your username for the GitHub. After entering the username you have to provide password ,for that you have to enter your Account Token you generated. Now copy-paste the Account Token. For pasting just Right click for a single time on the terminal (you won't be able to see your password).

- **ls**

```
ubuntu@ip-172-31-34-76:~$ ls
MyRepo2
```

- **cd myRepoV1/**

```
ubuntu@ip-172-31-34-76:~$ cd MyRepo2/
ubuntu@ip-172-31-34-76:~/MyRepo2$ ls
'New Text Document.txt'   index.js   package.json
ubuntu@ip-172-31-34-76:~/MyRepo2$
```

Now we enter into the directory, and we observe that we have all the files that we have in our Repository has been cloned in our directory in the server system.

- **npm install**

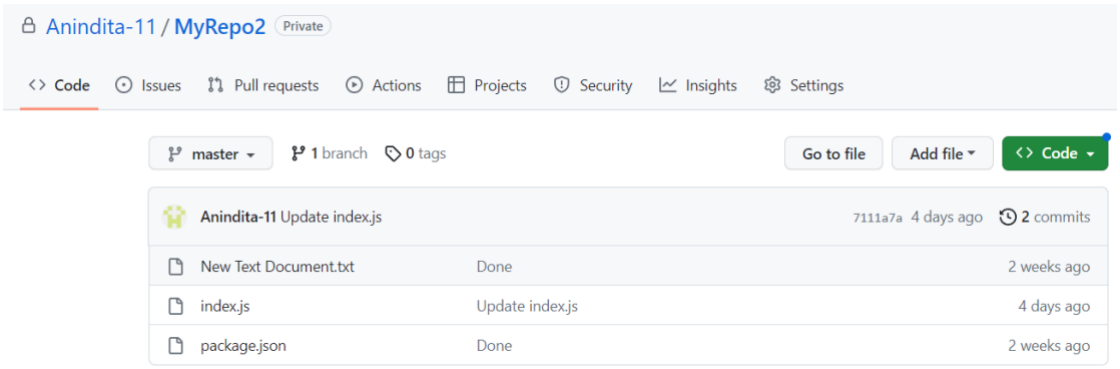
```
ubuntu@ip-172-31-34-76:~/MyRepo2$ npm install
npm WARN deprecated uuid@3.4.0: Please upgrade  to version 7 or higher.  Older versions may use Math.random() in certain circumstances, which is know
n to be problematic.  See https://v8.dev/blog/math-random for details.

added 258 packages, and audited 259 packages in 11s

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

found 0 vulnerabilities
npm notice
npm notice New minor version of npm available! 9.5.0 -> 9.6.4
npm notice Changelog: https://github.com/npm/cli/releases/tag/v9.6.4
npm notice Run npm install -g npm@9.6.4 to update!
npm notice
ubuntu@ip-172-31-34-76:~/MyRepo2$
```

Step 8: Go back to your Repository in Github. Open your “index.js” file.



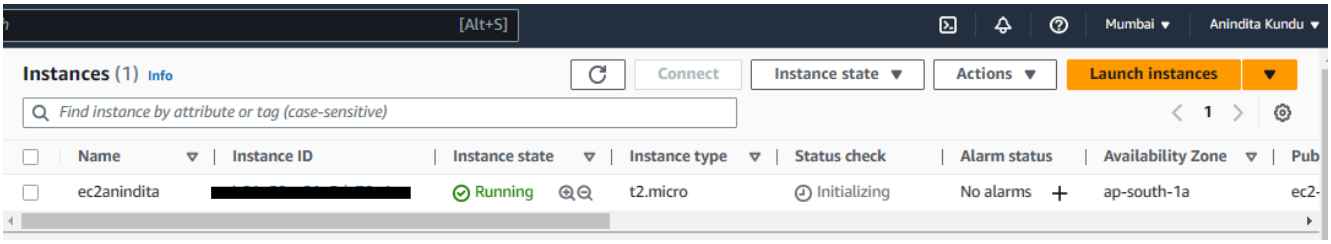
Step 9: Check the port no. specified in the program. It is specified in the app.listen() method as the first parameter. Here it is ‘4000’. Copy or remember this no. as it is the port no. and will be required to connect to our website.

11 lines (9 sloc) | 193 Bytes

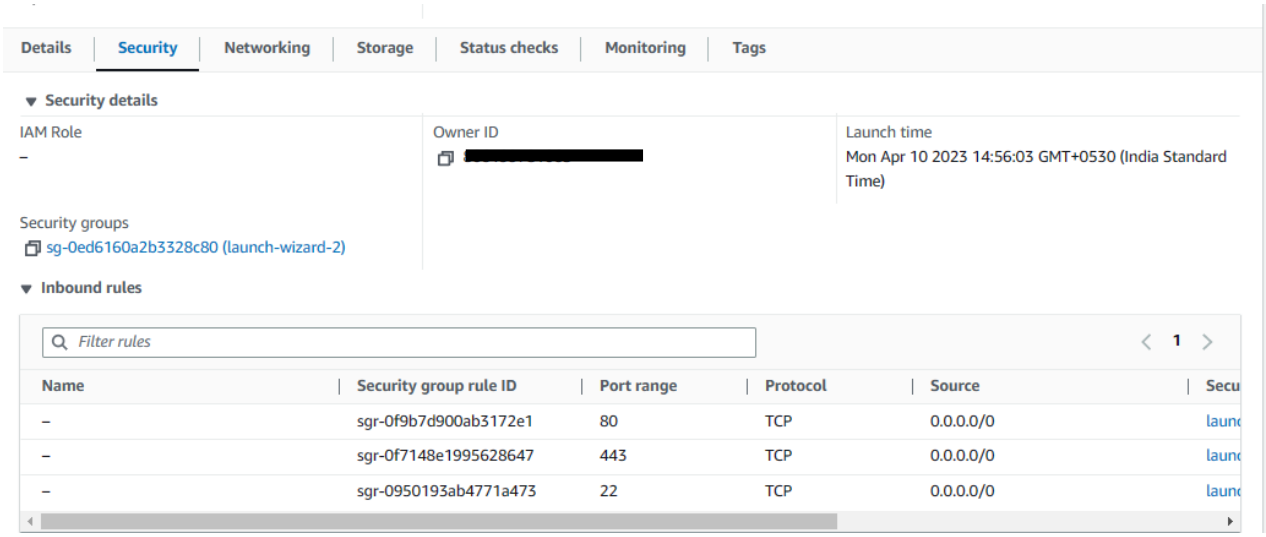
```
1  const express = require('express')
2  const app = express()
3
4  app.get('/', function (req, res) {
5    res.send('Hello,Anindita Here')
6  })
7
8  app.listen(4000, ()=>{
9      console.log("Started server");
10 }
11 )
```

We have to add this port no. to our EC2 instance security group rule otherwise we won’t be able to access the website from anywhere.

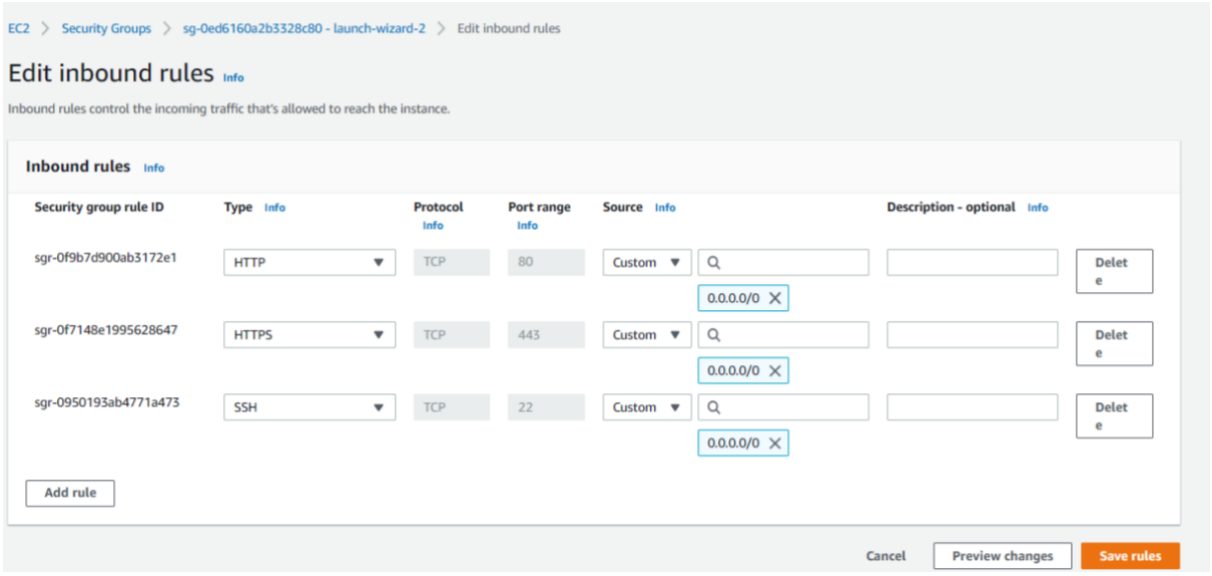
Step 10: Now go back to your AWS EC2 instances page. Click on the instance that is being used.



Step 11: Scroll down until you find a section bar where by default the details option is selected. Select the Security option. Then click on the security groups link under security groups.



Step 12: Then click on Edit Inbound Rules button. Click on the **Add Rule** button. A new Row will be generated. Let the type remain Custom TCP. Under Port Range write your Port no. you want to open. In this case we have 4000 port no. as we found out earlier in our index.js code. Next in source click on the search box and the first option with value **0.0.0.0/0** should be selected. Now click on the save rules button.



We have successfully added the Port No. to our Inbound rules. Now we can access our website. But first we need to start our server.

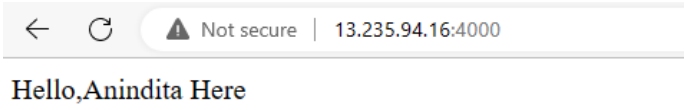
Step 13: We return back to the terminal and type:-

- **node index.js**

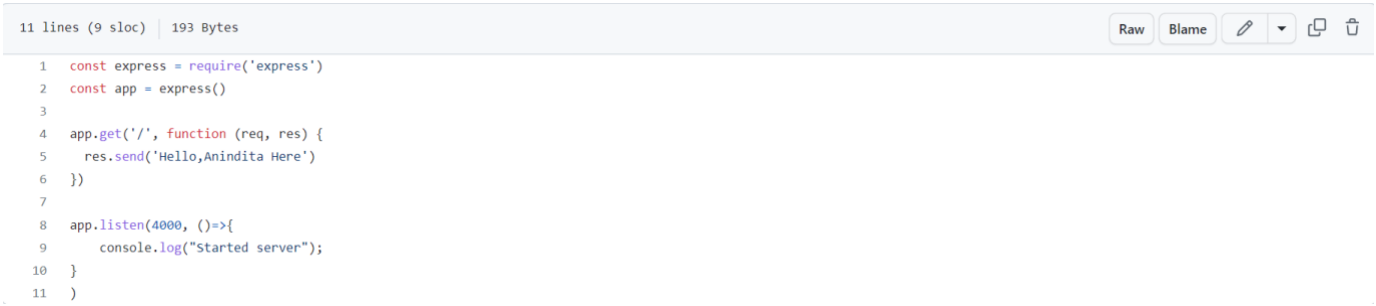
Our server has started and it is also reflected by the terminal prompt. Now to check we need to open another browser and type in the IPv4 address of our EC2 server to access our website**.**

Step 14: Now copy the IPv4 address of your EC2 server and paste it in another browser. But before pressing Enter add a colon (:) and then mention the Port No. mentioned in the index.js file. For our case it is 4000.

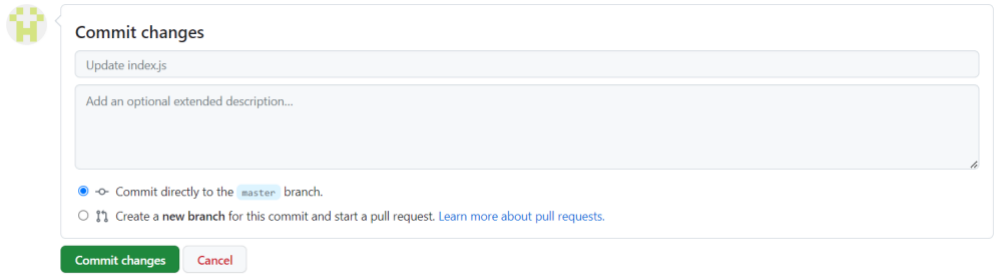
Now press Enter to load the website.



Step 15: We have successfully deployed our project from GitHub to our EC2 server. Now if we want to change something in our project or file or code we will follow these steps: Suppose we want to modify the displayed message. So open the **index.js** file in GitHub. After opening we click on the pen icon on the right corner side of the code viewer.



Step 16: We then modiify the string passed through the res.send() method. Now after editing, scroll-down and click the **Commit changes** button.Our changes have finally been commited in our Repository in GitHub.



Step 17: Now type:-

- **git pull**

(Enter the username when asked)
(Enter your account Token as your Password when asked for password)
(Right click once to paste, then press Enter)
Now we have to restart the server.

- **node index.js**

(We restarted the server)

Step 18: We now have to Refresh our browser where we have our website open. The changes have been successfully reflected. We have successfully completed our task of Deploying our project from GitHub to our EC2 server.