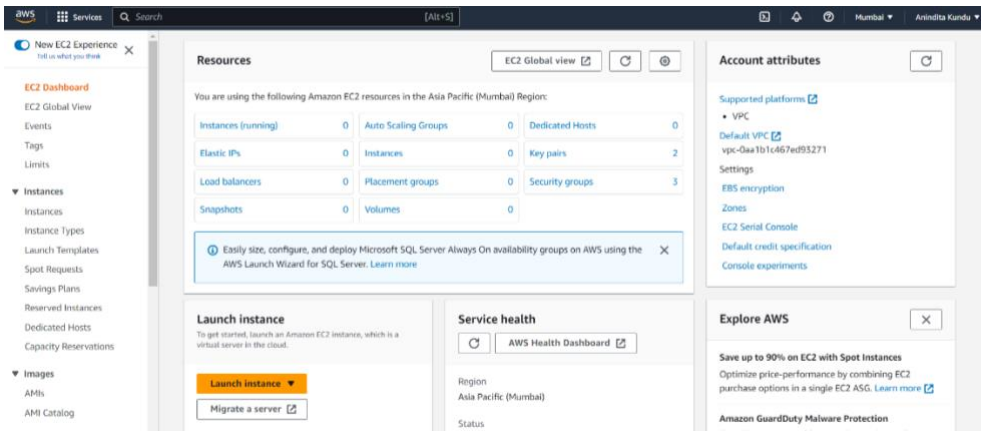
ASSIGNMENT – 10

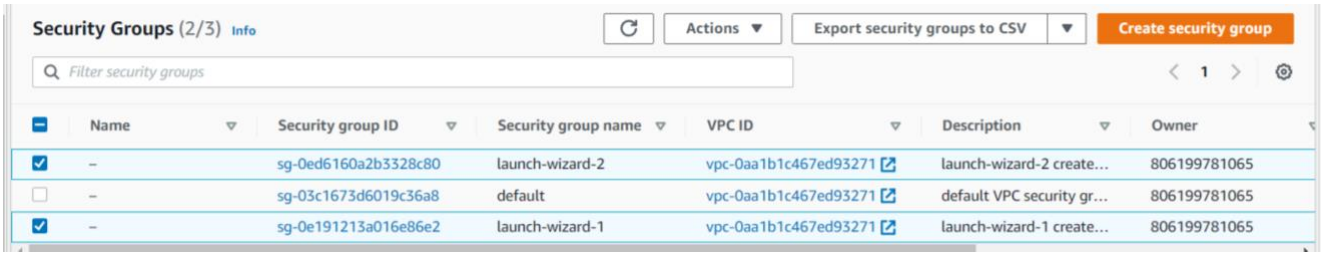
Problem Statement: Deploy project from GitHub to EC2 by creating new security group and user data.

Procedure:

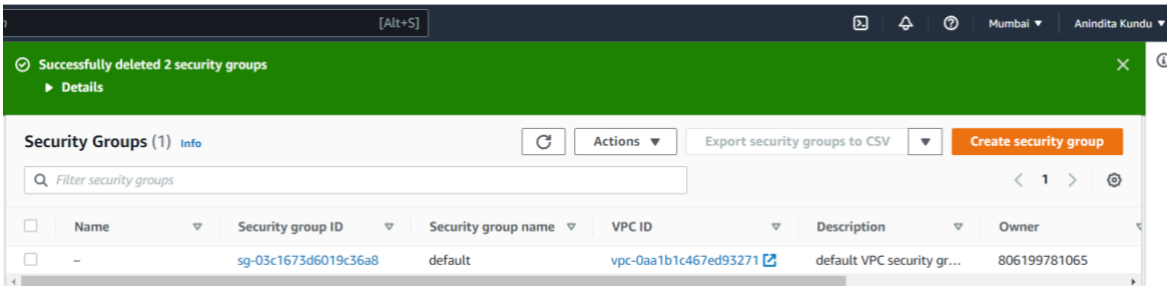
Step 1: Sign in to your AWS account. Go to your EC2 dashboard. Scroll down and Click on Security Groups option on the left side nav bar under Network & Security option.



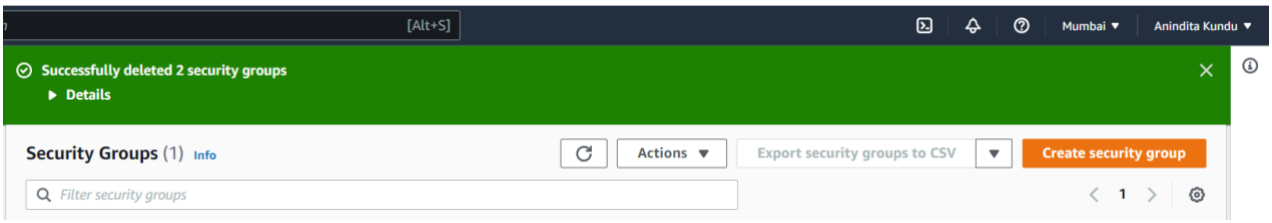
Step 2: Select all the Security Groups other than the one named "default".



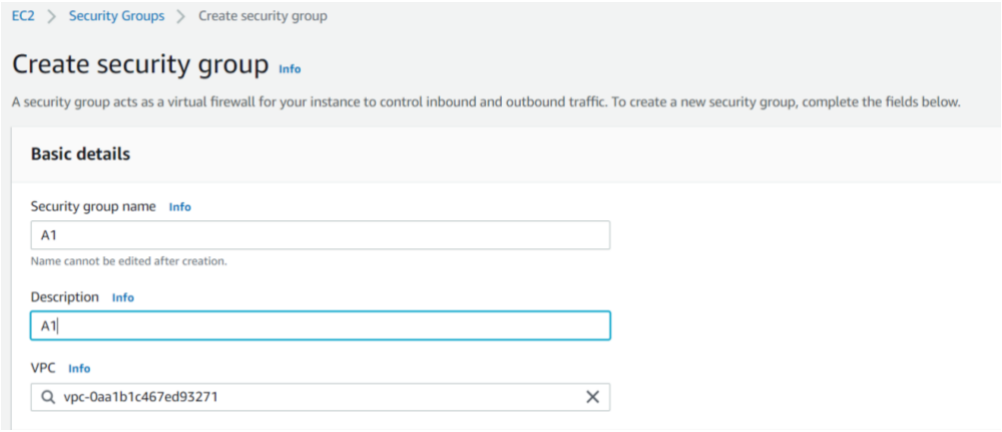
Step 3: Then Click on the Actions button.Scroll-Down the dropdown list until you find the "delete all security groups" option. Click on it.Now only the "default" security group remains and we keep it that way.



Step 4: Now click on the "Create Security Group" button.



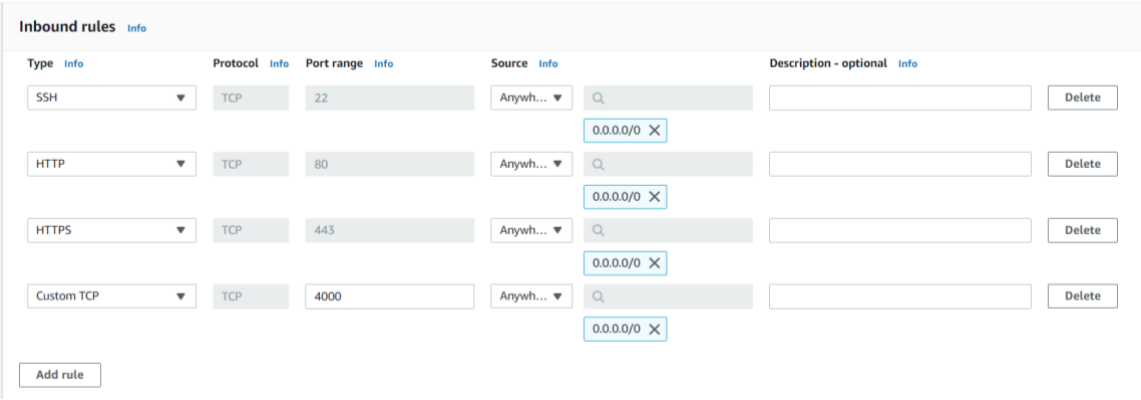
Step 5: Now start by giving a name to the security group and giving its description (anything). Let the VPC remain unchanged.



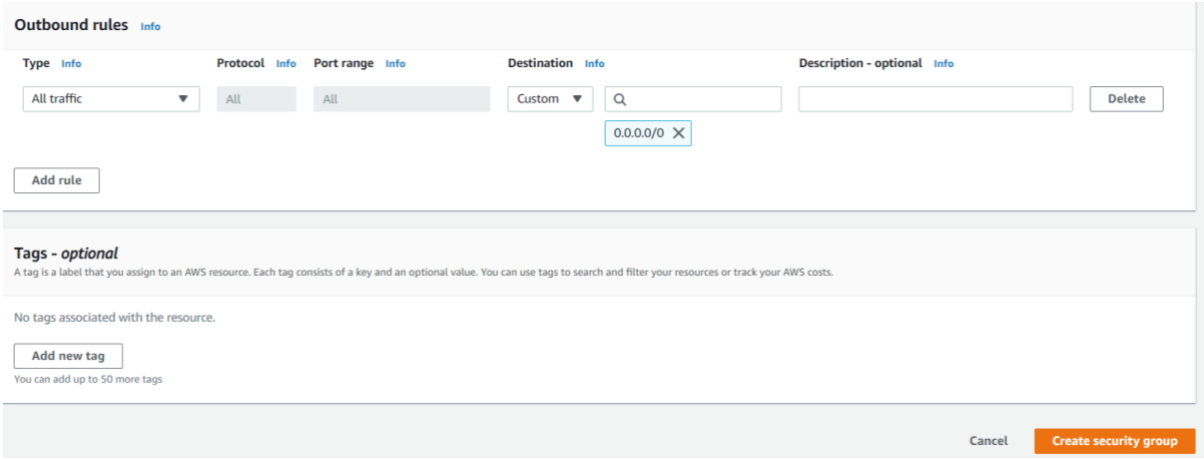
Step 6: Next, we will add Inbound Rules. Start adding by clicking the Add rule button. These include:

a) SSH
b) HTTP
c) HTTPS
d) Custom TCP

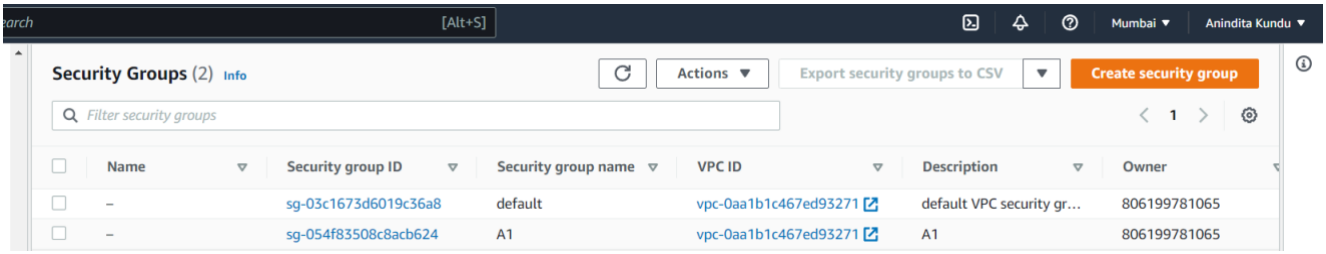
The last one with custom TCP has a specific port range that we require to connect to our project. It has been specified in our index.js file (refer to Ass9).



Step 7: Next outbound rules and all other sections remain unchanged. Now Click on the create security group button.

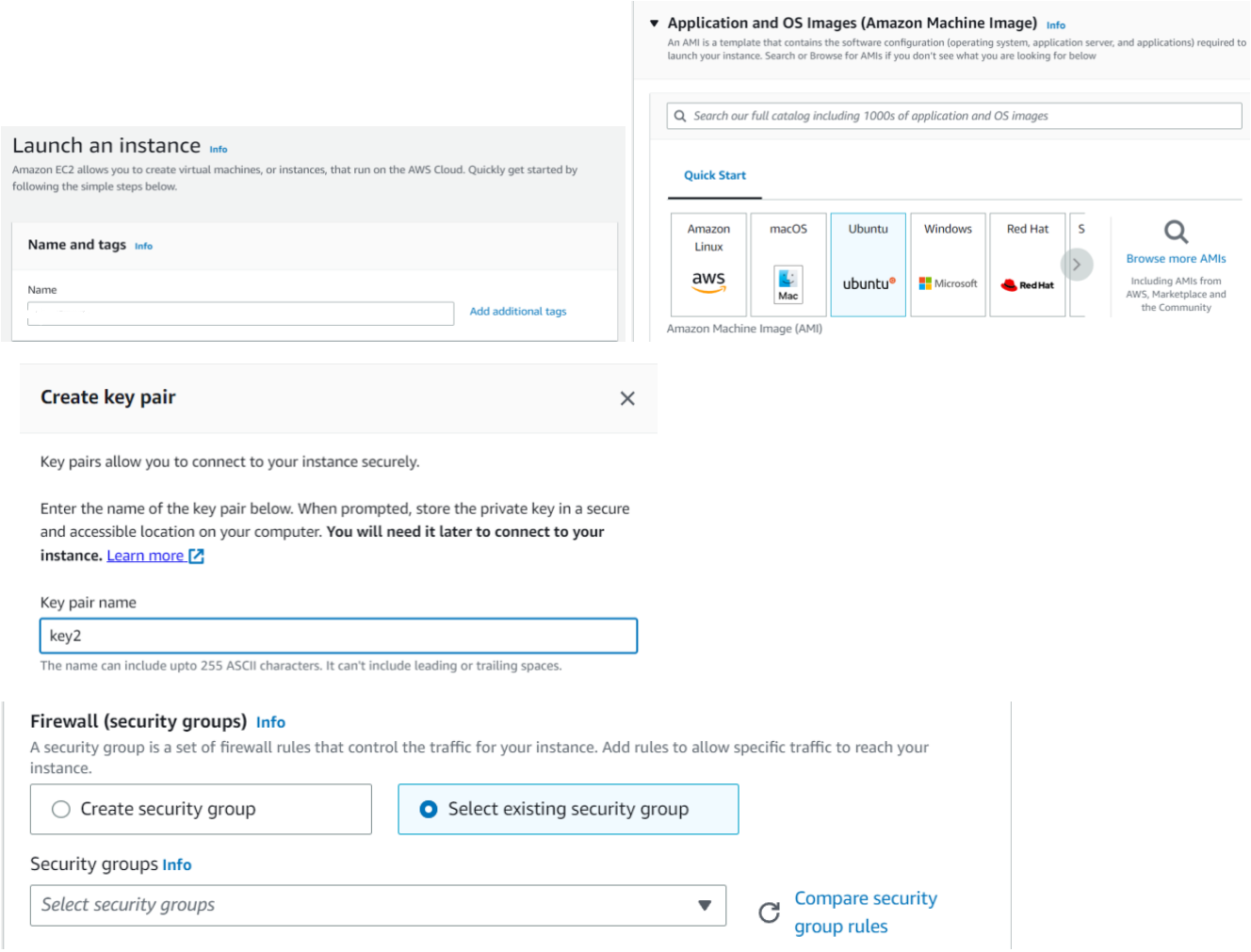


Step 8: Now go back to the security groups list and click on the security group ID of the newly created Security Group. After clicking we can view the inbound rules that we added during its creation.



Step 9: Now we go to the instances section from the left side nav bar.Now we Create a new EC2 instance. Click on the Launch Instance button.

Now, Give the name and Select Ubuntu as OS. Select a keypair or generate a new one if none is available. Then under Network settings select the Select Existing Security Group option.



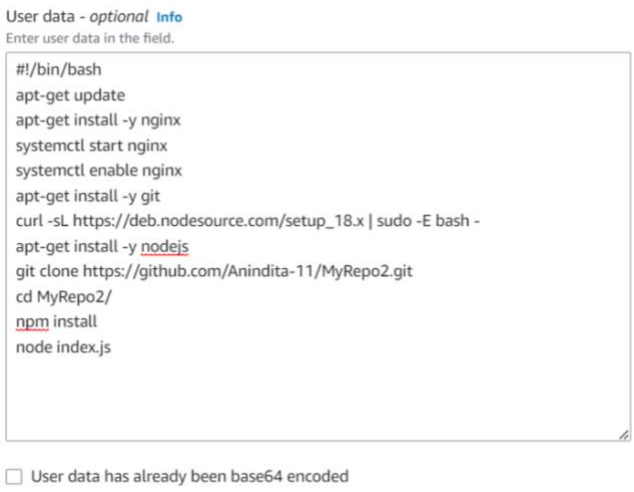
Now under the security groups dropdown menu select the one we just created.

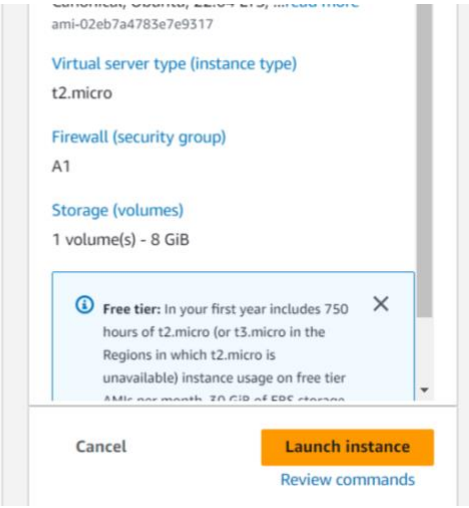
Now scroll down and click on the Advanced Details option. Then again scroll-down to the newly appeared sub-sections until you find User Data section.
Write the following commands in the given box. Remember this user data is given to execute the given commands once the server starts.

```
#!/bin/bash
apt-get update
apt-get install -y nginx
systemctl start nginx
systemctl enable nginx
apt-get install -y git
curl -sL https://deb.nodesource.com/setup_18.x | sudo -E bash -
apt-get install -y nodejs
```

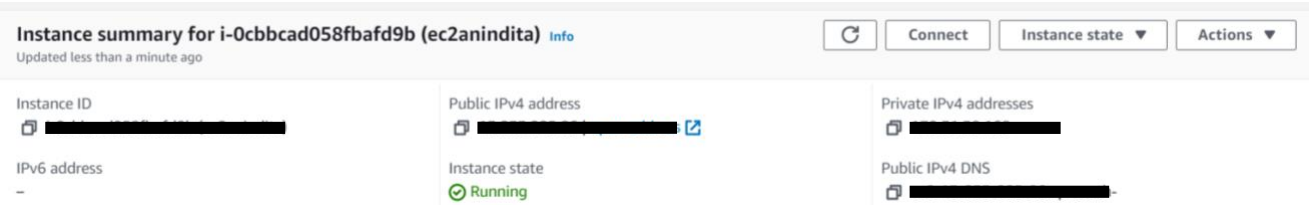
We have created a private repository in GitHub. So, whenever we run the git clone command it asks for our username and password. Hence this cannot be executed directly through our User Data instructions. We have to connect manually and enter all commands starting from the git clone command.

- Now we click on the launch instance button.





Step 10: Now we Click on the 'Instance Id' link of our newly created server in our Instances list. Now click on the connect button



Again, click on the connect button. After this anew Tab will open with a Bash Terminal that is of our remote EC2 server.

Step 11: after opening the Terminal, we can type all our required commands that we used to type in a similar terminal by connecting to our remote server through our Bitvise SSH client software in our previous assignments.

```
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status



The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
```

Step 12: Now type the following commands in the terminal:-

- git clone **your GitHub Repository URL**
- your Username of GitHub will be asked.
- your account Token as your Password will be asked.

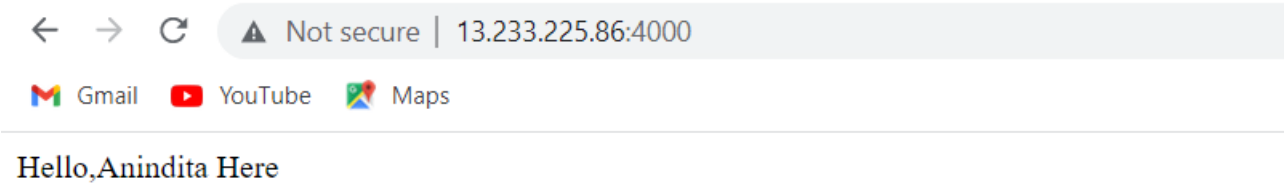
```
ubuntu@ip-172-31-38-168:~$ git clone https://github.com/Anindita-11/MyRepo2.git
Cloning into 'MyRepo2'...
Username for 'https://github.com': Anindita-11
Password for 'https://Anindita-11@github.com':
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 8 (delta 1), reused 4 (delta 0), pack-reused 0
Receiving objects: 100% (8/8), done.
Resolving deltas: 100% (1/1), done.
```

- cd YourRepositoryname
- npm install
- node index.js

```
ubuntu@ip-172-31-38-168:~$ cd MyRepo2/
ubuntu@ip-172-31-38-168:~/MyRepo2$ npm install
npm WARN deprecated uuid@3.4.0: Please upgrade  to version 7 or higher.  Older versions
ee 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! 9.5.1 -> 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-172-31-38-168:~/MyRepo2$ node index.js
Started server
^C
ubuntu@ip-172-31-38-168:~/MyRepo2$
```

Step 13: Now copy and paste the Public IPv4 address of your EC2 instance in another browser.

Step 14: Now append the port no. 4000 (for our case) to the IP address in the browser with a ":" sign.



We have successfully Deployed a project from GitHub to EC2 by creating a new Security group and User Data.