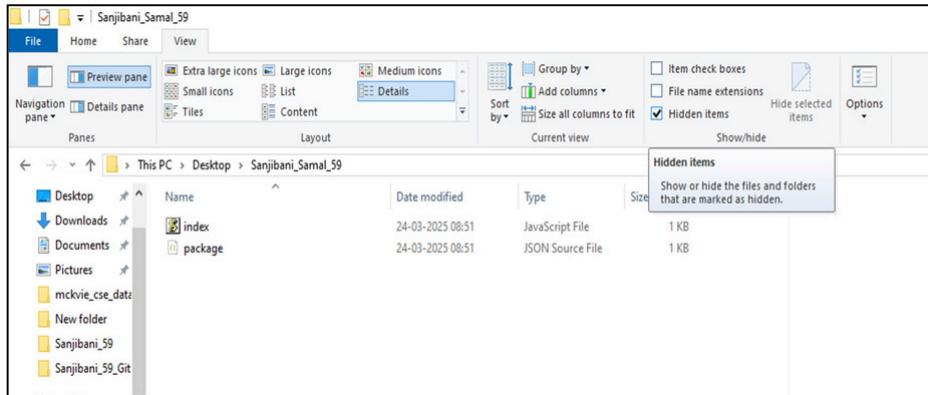


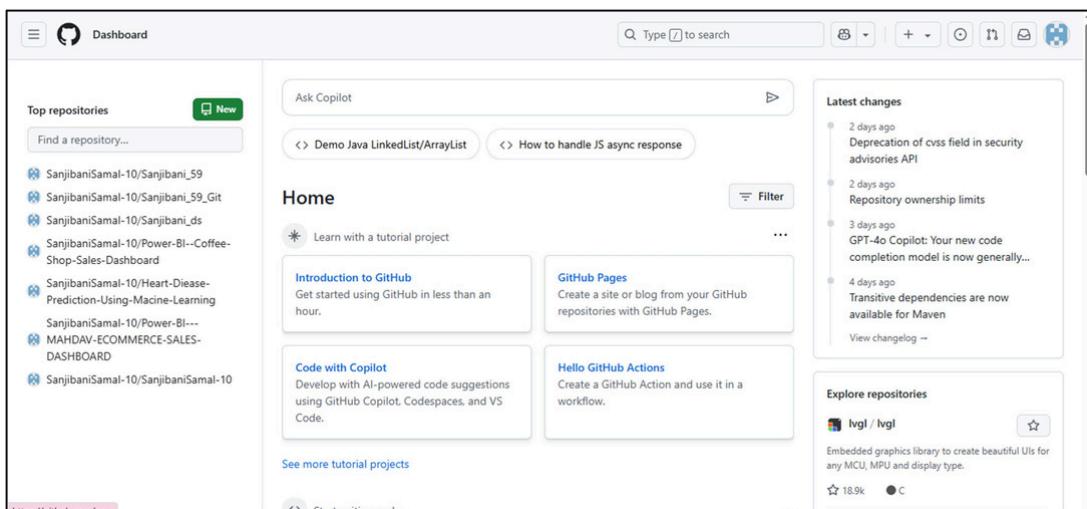
ASSIGNMENT NO :- 10

TITLE :- Deploy a project from GitHub to EC2 by creating a new Security group and user data .

STEP 01 :- Create two files Index.js and package.json and store it in a folder here the folder name is 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 * SanjibaniSamal-10 / **Repository name *** Sanjibani_Samal_59
Sanjibani_Samal_59 is available.

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

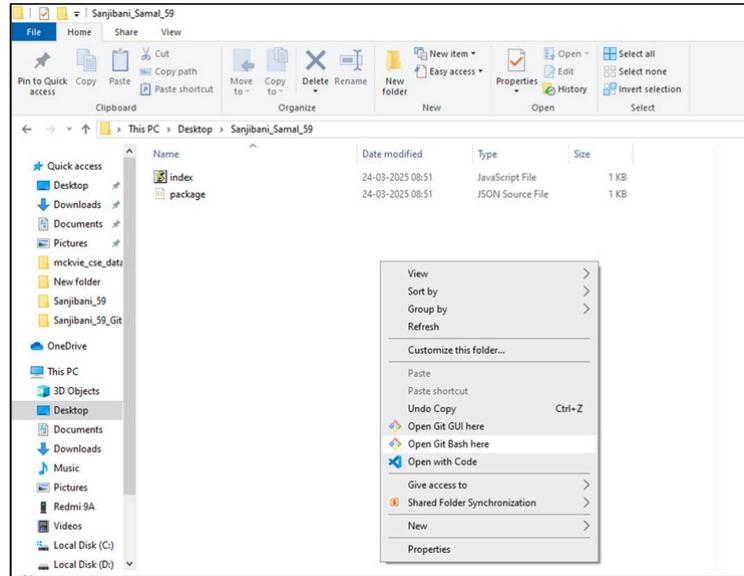
or https://github.com/SanjibaniSamal-10/Sanjibani_Samal_59.git

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.

The screenshot shows a GitHub repository named 'Sanjibani_Samal_59'. The 'Code' tab is selected. The master branch has 1 commit. The repository has 0 stars, 1 watching, and 0 forks. The README section is empty with a 'Add a README' button.

STEP 08:- Click on index.js

The screenshot shows the same GitHub repository page. The 'index.js' file is now highlighted in blue in the file list. The repository details and README section remain the same.

STEP 09:- Click on Edit the file and edit the file

The screenshot shows the GitHub code editor for the 'index.js' file. The code is as follows:

```
1 const express = require('express')
2 const app = express()
3
4 app.get('/', function (req, res) {
5   res.send('Hello MCKV')
6 })
7
8 app.get('/courses', function (req, res) {
9   res.send('Print all courses')
10 })
11
12 app.listen(4000, ()=>{
13   console.log("Started server");
14 })
15 )
```

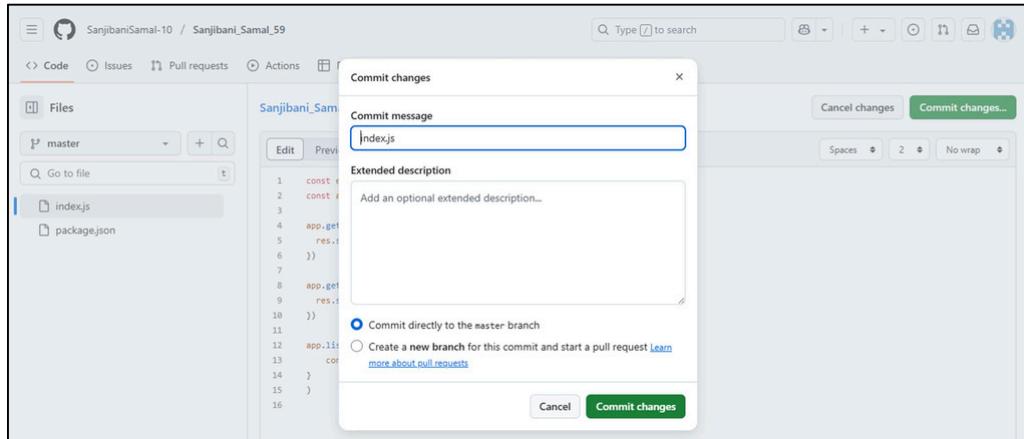
STEP 10 :- Click on Commit Changes and Give the Commit message and click on Commit change

```
const express = require('express')
const app = express()

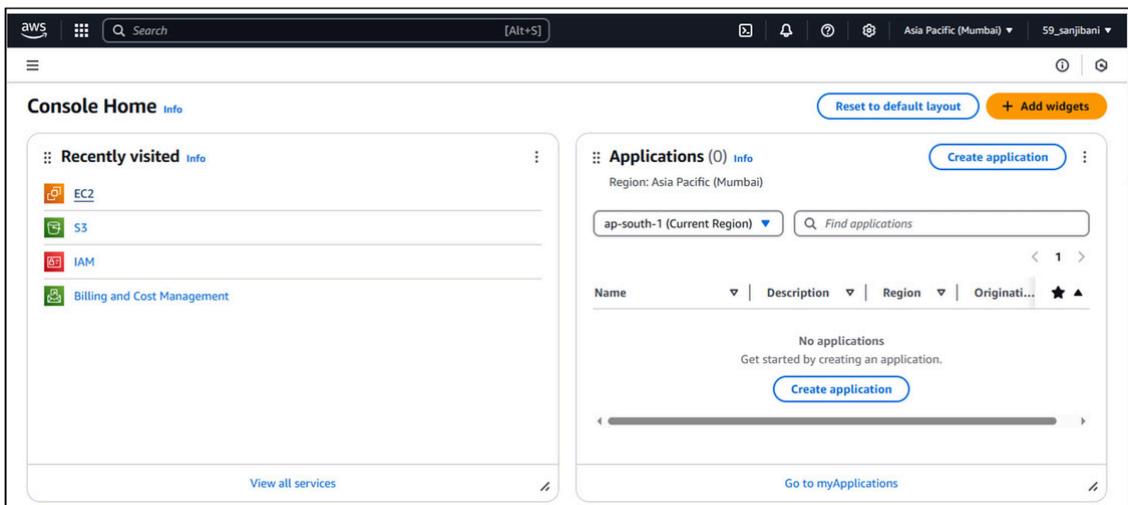
app.get('/', function (req, res) {
  res.send('Hello MCKVIAN SANJIBANI !!!')
})

app.get('/courses', function (req, res) {
  res.send('Print all courses')
})

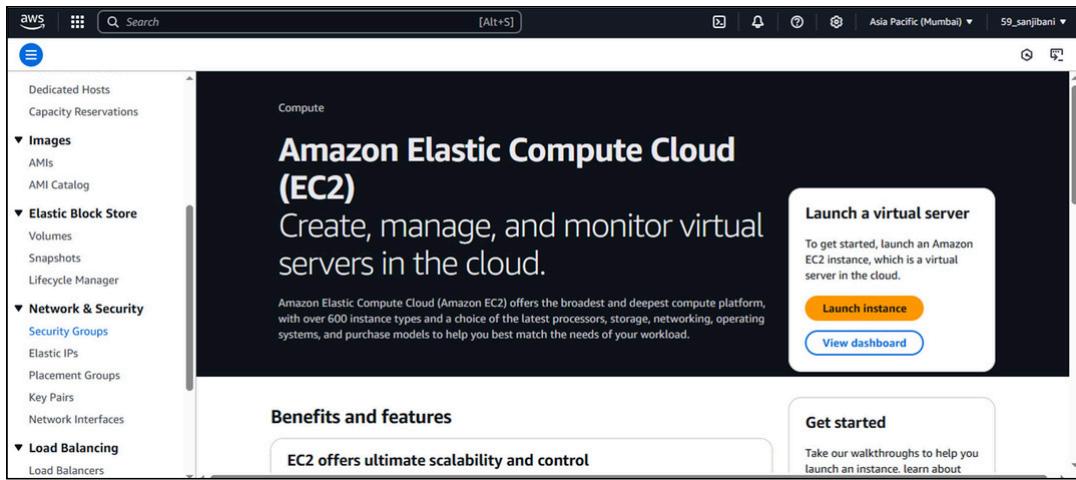
app.listen(4000, ()=>{
  console.log("Started server")
})
```



STEP 11 :- Login to your AWS Account and select EC2.



STEP 12 :- Goto the Security Groups



STEP 13 :- Delete all the Security Groups except the Default.

A screenshot of the AWS Security Groups list. The sidebar shows the same navigation as the previous screenshot. The main table lists four security groups: sg-0501bd6fd6b3c5d6d, sg-0c883af72e5b40bf9, sg-06d6eb857b20c088c, and sg-066f0c5a134ddc460. A context menu is open over the second group, showing options like "View details", "Edit inbound rules", "Edit outbound rules", "Manage tags", "Manage stale rules", "Copy to new security group", "Share security group", and "Delete security groups". The "Delete security groups" option is highlighted.

A screenshot of the AWS Security Groups list with a confirmation dialog box. The dialog asks, "Are you sure that you want to delete these security groups?" and lists the three selected groups: sg-0c883af72e5b40bf9 - launch-wizard-1, sg-06d6eb857b20c088c - launch-wizard-2, and sg-066f0c5a134ddc460 - ss. Below the list, it says "To confirm deletion, enter delete below." A text input field contains the word "delete". At the bottom right of the dialog are "Cancel" and "Delete" buttons.

>> All the security group Successfully Deleted

The screenshot shows the AWS VPC Security Groups page. A green success message at the top states "Successfully deleted 3 security groups". Below it, a table lists one security group: "sg-0501bd6fd6b3c5d6d" with "default" as the name and "vpc-064e179da3f6ea9d8" as the VPC ID. The left sidebar includes sections for Dedicated Hosts, Capacity Reservations, Images (AMIs, AMI Catalog), Elastic Block Store (Volumes, Snapshots, Lifecycle Manager), Network & Security (Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces), and Load Balancing (Load Balancers). The bottom right corner contains copyright information and links for Privacy, Terms, and Cookie preferences.

STEP 14 :- After the Successful deletion of all Security Groups click on Create Security Groups

This screenshot is identical to the previous one, showing the AWS VPC Security Groups page. It displays a single security group named "sg-0501bd6fd6b3c5d6d" with "default" as its name and "vpc-064e179da3f6ea9d8" as its VPC ID. The interface and sidebar are the same, indicating no further changes have occurred since the previous step.

STEP 15 :- Give the Security Group Name and Description and the click on Add Rules of Inbound Rules

Basic details

Security group name [Info](#)
sanjibani
Name cannot be edited after creation.

Description [Info](#)
sanjibani

VPC [Info](#)
vpc-064e179da3f6ea9d8

Inbound rules [Info](#)
This security group has no inbound rules.
[Add rule](#)

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

>> Add SSH , HTTP , HTTPS and CUSTOM TCP Type Rules as Given below

Inbound rules [Info](#)

Type	Protocol	Port range	Source	Description - optional
SSH	TCP	22	Any... Info	0.0.0.0/0 Delete
HTTP	TCP	80	Any... Info	0.0.0.0/0 Delete
HTTPS	TCP	443	Any... Info	0.0.0.0/0 Delete
Custom TCP	TCP	4000	Any... Info	0.0.0.0/0 Delete
Custom TCP	TCP	4000	Any... Info	0.0.0.0/0 Delete

[Add rule](#)

⚠️ 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. [X](#)

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

>> Click on CREATE Security Group

All traffic [Delete](#)

[Add rule](#)

⚠️ Rules with destination of 0.0.0.0/0 or ::/0 allow your instances to send traffic to any IPv4 or IPv6 address. We recommend setting security group rules to be more restrictive and to only allow traffic to specific known IP addresses. [X](#)

Tags - optional
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.
No tags associated with the resource.
[Add new tag](#)
You can add up to 50 more tags

Create security group

Swastik Ghosh
B-TECH/CSE-DS/22/019

>> And the Security Group is Successfully created

The screenshot shows the AWS EC2 Security Groups page. A green success message at the top states: "Security group (sg-020bedda11e092c8d | sanjibani) was created successfully". Below this, the security group details are listed:

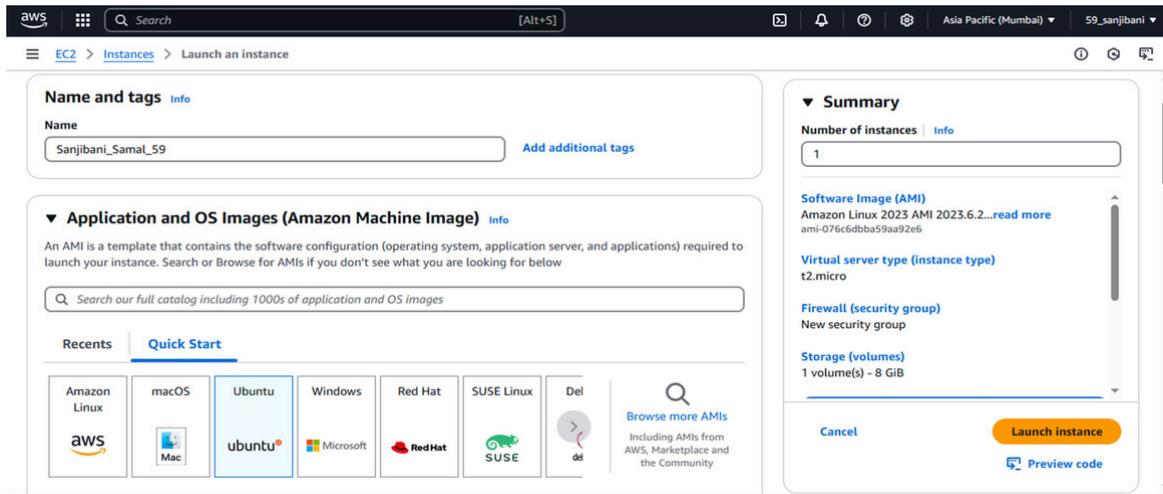
Security group name	Security group ID	Description	VPC ID
sanjibani	sg-020bedda11e092c8d	sanjibani	vpc-064e179da3f6ea9d8

Owner: 676206938169, Inbound rules count: 4 Permission entries, Outbound rules count: 1 Permission entry.

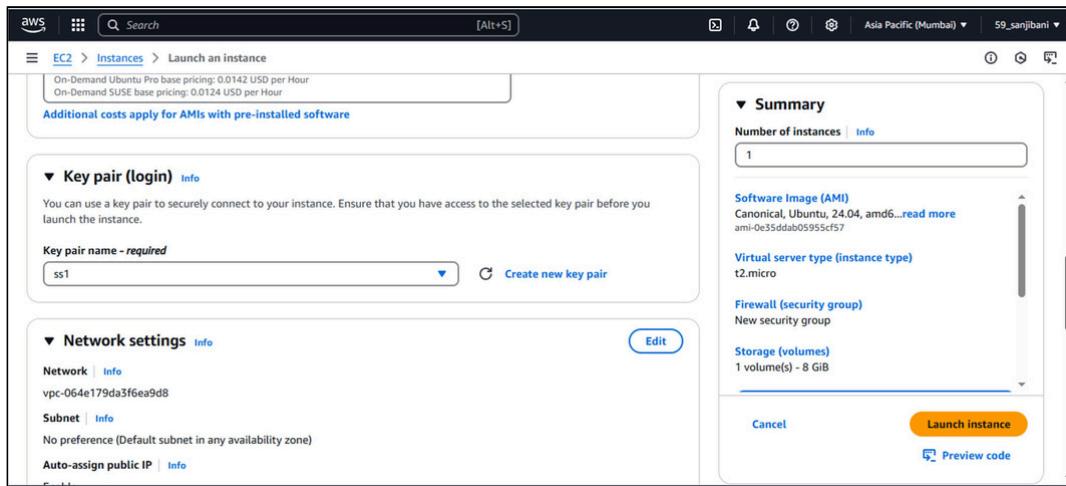
Below the details, there are tabs for Inbound rules, Outbound rules, Sharing - new, VPC associations - new, and Tags. The Inbound rules tab is selected, showing 4 rules. At the bottom right of the main content area, there are links for Manage tags and Edit inbound rules.

STEP 16 :- Click on Launch Instance and Give your instance name and then select Ubuntu and select a key

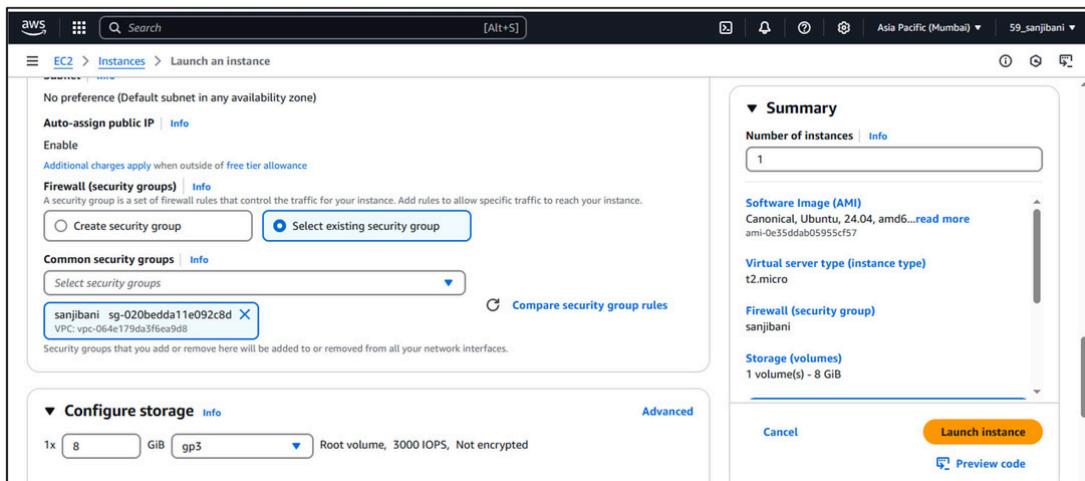
The screenshot shows the AWS EC2 Instances page. The left sidebar is expanded to show the Instances section. The main content area displays the "Instances Info" section, which is currently empty. It includes filters for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability. A prominent blue "Launch instances" button is located in the center of the page. Below the button, a placeholder text says "Select an instance".



>> Here I have Selected a existing Key ss1

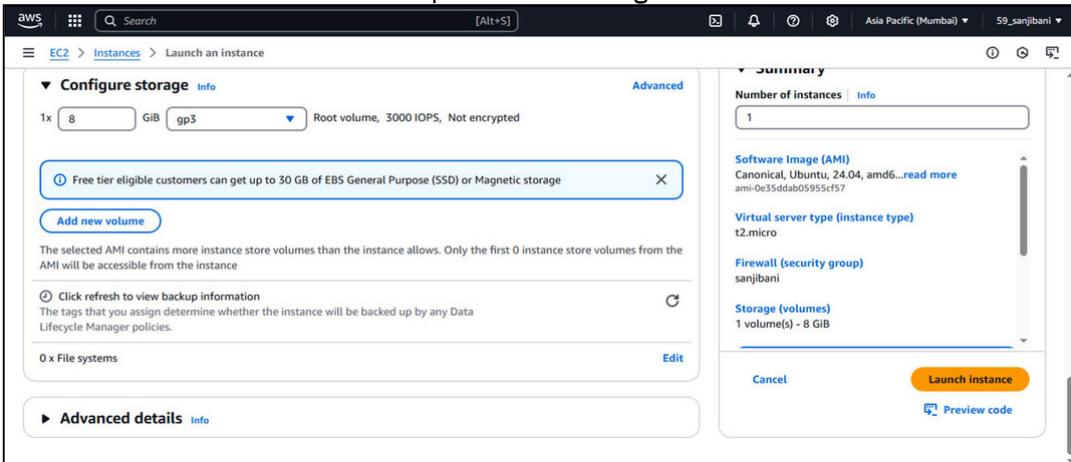


STEP 17 :- Then select Select existing security group and then choose your security group.

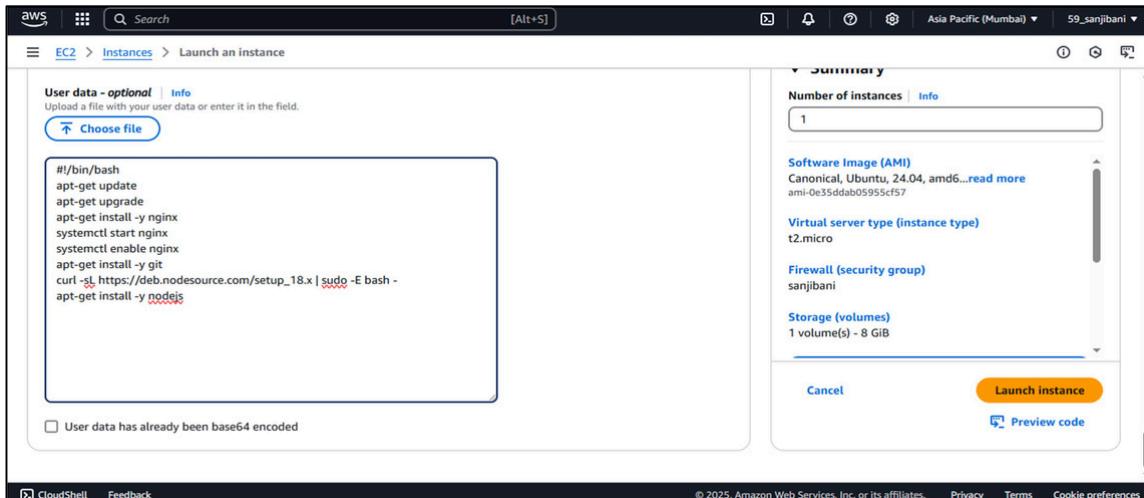


Swastik Ghosh
B-TECH/CSE-DS/22/019

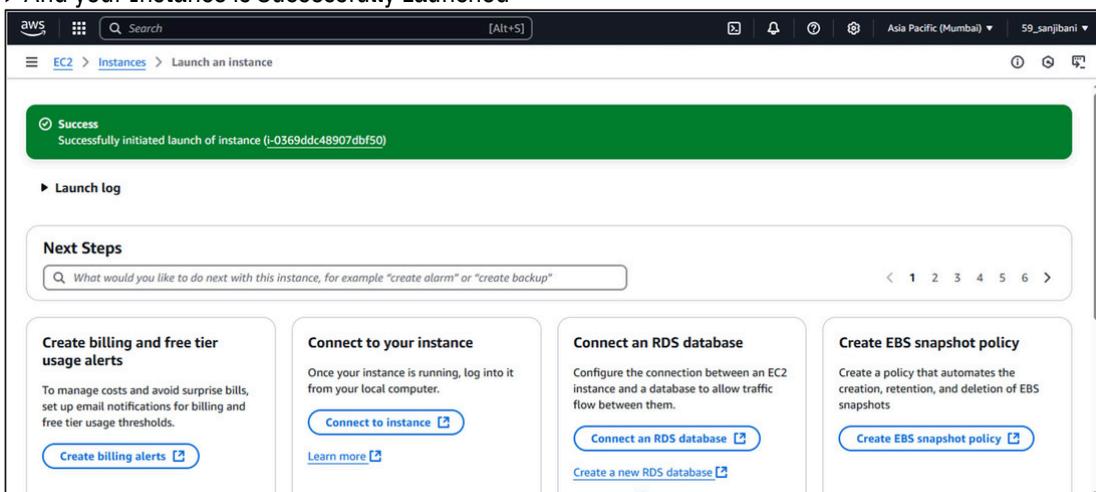
STEP 18 :- Then click on Advanced options and then go to User data



>> Write the following commands given below in User data and click on Launch Instance in Right hand bottom of page.



>> And your Instance is Successfully Launched



STEP 19 :- Click on the Instance ID and copy the Public IPv4 Address.

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with options like Dashboard, EC2 Global View, Events, Instances, Images, and Elastic Block Store. The main area displays an instance summary for 'i-0369ddc48907dbf50 (Sanjibani_Samal_59)'. The summary includes:

- Instance ID:** i-0369ddc48907dbf50
- Instance state:** Running
- Public IPv4 address:** 65.1.95.153 (with a 'copy' tooltip)
- Private IP DNS name (IPv4 only):** ip-172-31-3-2.ap-south-1.compute.internal
- Instance type:** t2.micro
- VPC ID:** vpc-064e179da3f6ea9d8

On the right, there are sections for Private IPv4 addresses (172.31.3.2), Public IPv4 DNS (ec2-65-1-95-153.ap-south-1.compute.amazonaws.com), and Elastic IP addresses (none).

STEP 20 :- Open the Bitvise SSH Client Server and paste the Public IPv4 Address in the Host As the key is the Existing one so all other features will remain same. And then click on Log in and open the Terminal

The screenshot shows two windows side-by-side. On the left is the Bitvise SSH Client 9.42 interface, showing a 'Default profile' configuration. The 'Server' section has 'Host' set to '65.1.95.153'. The 'Authentication' section has 'Username' set to 'ubuntu' and 'Client key' set to 'Global 2'. On the right is the AWS EC2 instance details page for 'i-0369ddc48907dbf50 (Sanjibani_Samal_59)'. It shows the same Public IPv4 address (65.1.95.153) and other instance details.

STEP 21 :- Goto to your GitHub Account Copy you Repository URL by clicking on Code

STEP 22 :- Then write the command in terminal and run it

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

```
ubuntu@ip-172-31-3-2:~$ node -v
v18.20.8
ubuntu@ip-172-31-3-2:~$ git clone https://github.com/SanjibaniSamal-10/Sanjibani_Samal_59.git
Cloning into 'Sanjibani_Samal_59'...
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 7 (delta 0), reused 3 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (7/7), done.
Resolving deltas: 100% (1/1), done.
ubuntu@ip-172-31-3-2:~$ cd Sanjibani_Samal_59
ubuntu@ip-172-31-3-2:~/Sanjibani_Samal_59$ ls
index.js  package.json
ubuntu@ip-172-31-3-2:~/Sanjibani_Samal_59$ npm install
added 227 packages, and audited 228 packages in 14s
```

STEP 23 :- Write node index.js in terminal to Start the Server

```
npm notice
ubuntu@ip-172-31-3-2:~/Sanjibani_Samal_59$ ^C
ubuntu@ip-172-31-3-2:~/Sanjibani_Samal_59$ node index.js
Started server
```

STEP 24 :- 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 with a specific instance selected. The instance summary includes:

- Instance ID: i-0369ddc48907dbf50
- IPv6 address: -
- Hostname type: IP name: ip-172-31-3-2.ap-south-1.compute.internal
- Answer private resource DNS name: IPv4 (A)
- Auto-assigned IP address: 65.1.95.153 [Public IP]
- Private IP DNS name (IPv4 only): ip-172-31-3-2.ap-south-1.compute.internal
- Instance state: Running
- Instance type: t2.micro
- VPC ID: vpc-064e179da3f6ea9d8

A context menu is open in a browser window, with the "Public IPv4 address copied" option highlighted.

