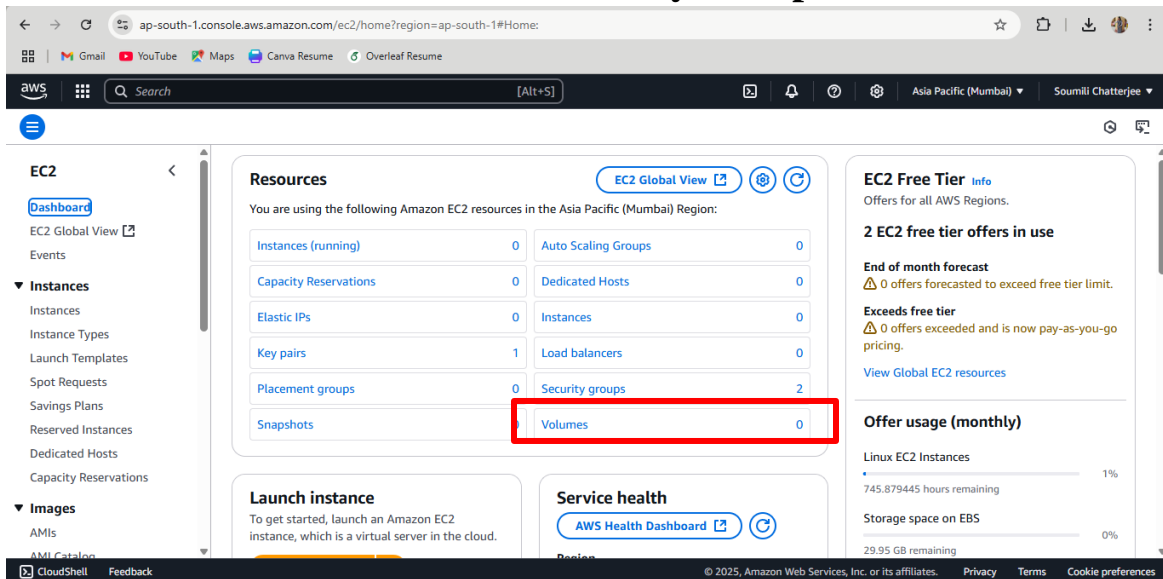


## Assignment No. – 10

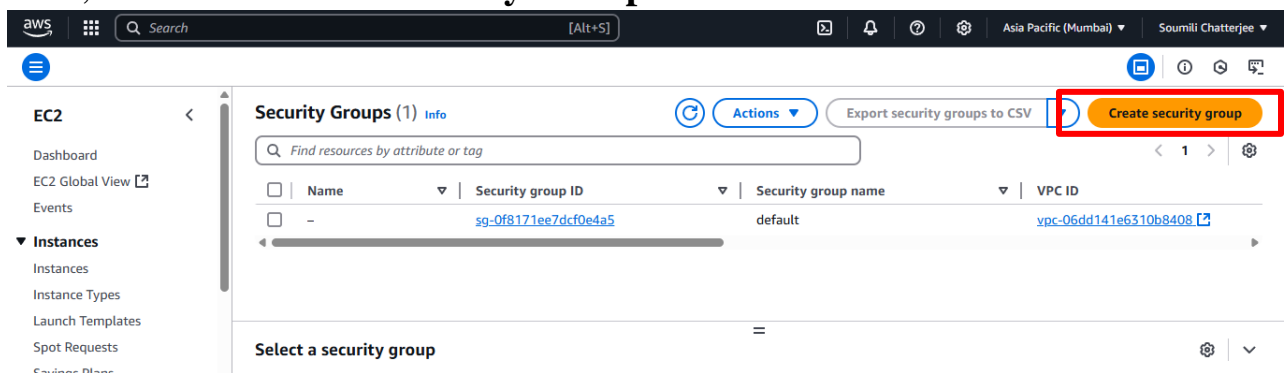
**Problem Statement:** Deploy a project from GitHub to EC2 by creating a new security group and user data.

### **Procedure:**

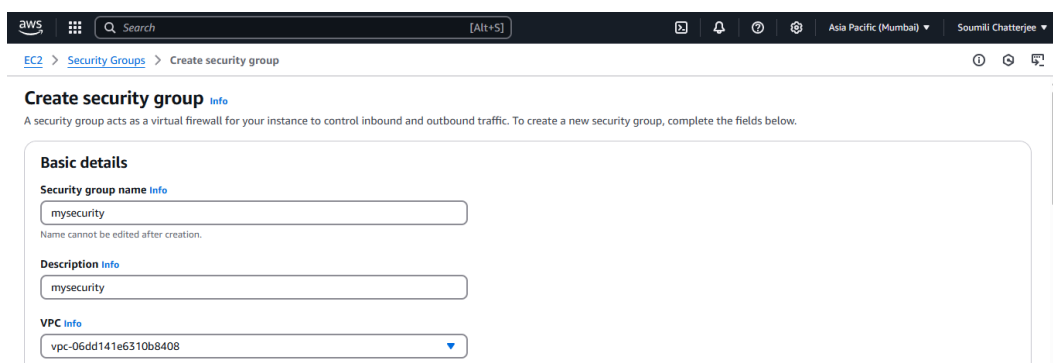
1. **Sign In** to the AWS account.
2. Go to **EC2 dashboard**.
3. Under **Resources** section click on **Security Groups**.



4. Select all the **Security Groups** other than the one named “**default**”. Click on **Actions** button. Scroll down and click on “**delete all security groups**”.
5. Next, click on “**Create Security Group**” button.



6. Now, give a name to the security group and write the same in the description below.



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

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

The first one with custom TCP has a specific port range(i.e.,4000) which is required to connect the project.

8. All other sections remain unchanged. Next, click on the **Create security group** button.

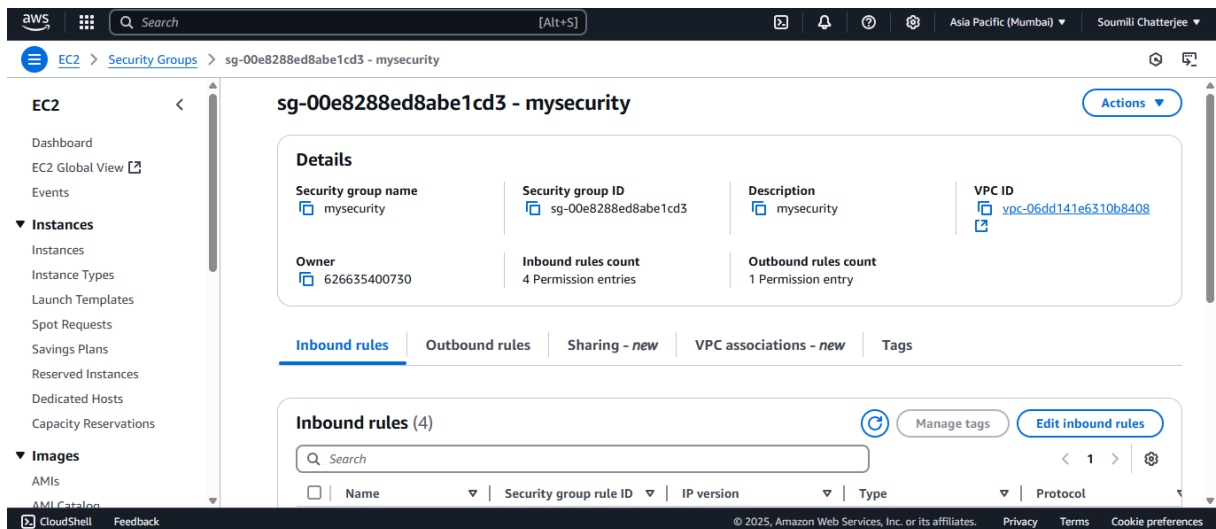
Rules with destination of 0.0.0.0/0 or ::/0 allow your instances to send traffic to any IPv4 or IPv6 address. We recommend setting security group rules to be more restrictive and to only allow traffic to specific known IP addresses.

**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.

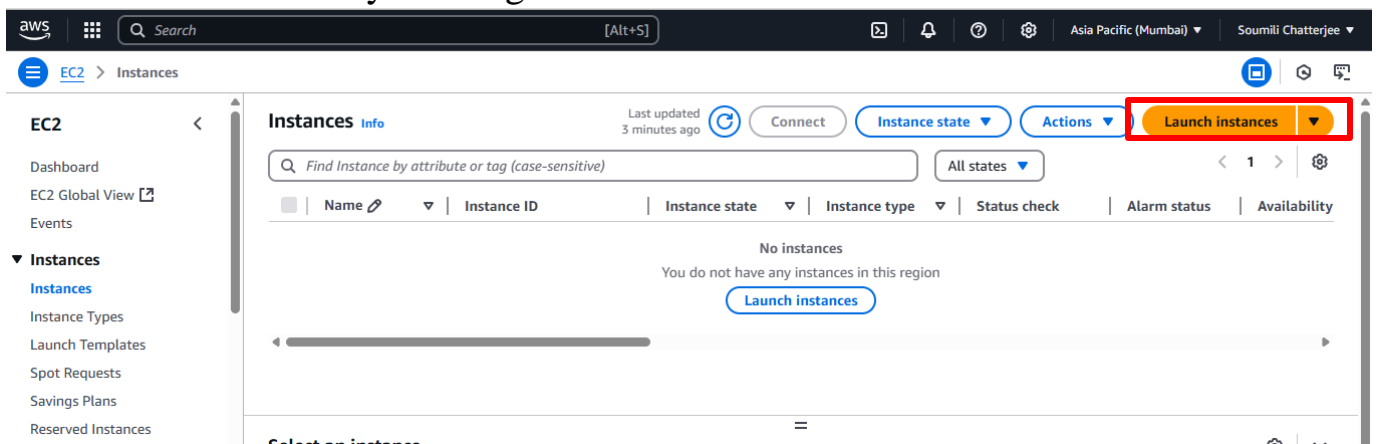
**Create security group**

9. Now. go back to the security groups list and click on the **security group ID** of the newly created Security Group.

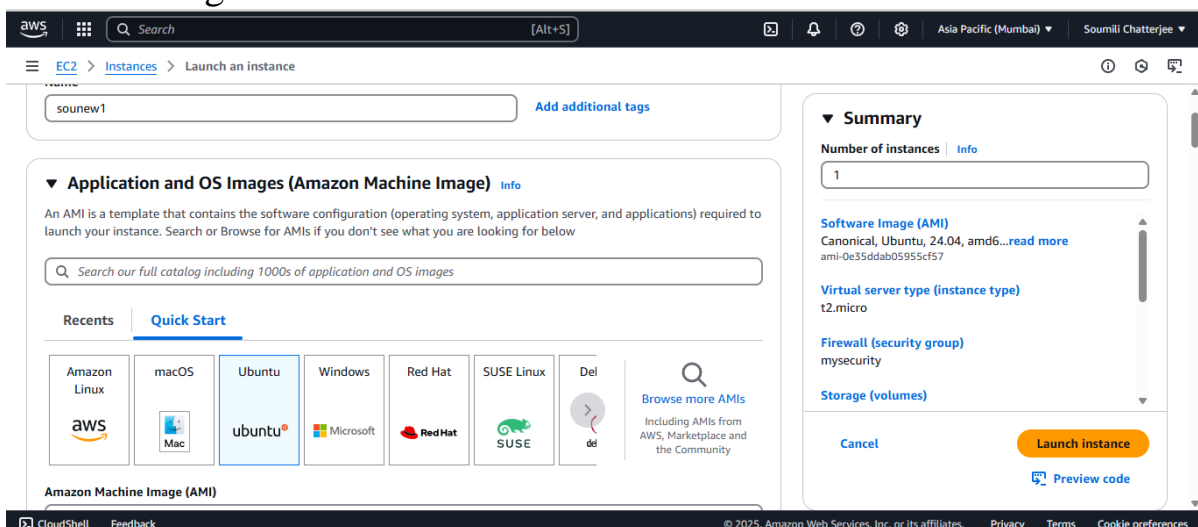
Name	Security group ID	Security group name	VPC ID
-	sg-00e8288ed8abe1cd3	mysecurity	vpc-06dd141e6310b8408
-	sg-0f8171ee7dcf0e4a5	default	vpc-06dd141e6310b8408



10. Now, we go to the **instances** section from the left side navigation bar. We create a new **EC2 instance** by clicking on the **Launch Instance** button.



11. Now, give a **name**, select **Ubuntu** as **OS**, select a **keypair** or generate a new one if none is available, then under **Network settings** select “**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 and scroll-down to the newly appeared sub-sections until you find **User Data** section. Write the commands in the given box.




### ▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

key1

 [Create new key pair](#)

### Subnet [Info](#)

No preference (Default subnet in any availability zone)

### Auto-assign public IP [Info](#)

Enable

[Additional charges apply](#) when outside of [free tier allowance](#)

### Firewall (security groups) [Info](#)


A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☐ Create security group

☒ Select existing security group

### Common security groups [Info](#)

Select security groups

mysecurity sg-00e8288ed8abe1cd3   
VPC: vpc-06dd141e6310b8408

 [Compare security group rules](#)

Security groups that you add or remove here will be added to or removed from all your network interfaces.

### User data - *optional* [Info](#)

Upload a file with your user data or enter it in the field.

 [Choose file](#)

```
#!/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_16.x|sudo -E bash -
apt-get install -y nodejs
git clone http://github.com/sudip7407/Repo1.git
cd Repo1
npm install
node index.js
```



12. Now, we click on **launch instance** button.

13. Next, we click on the '**Instance Id**' link of our newly created server in our Instances list.

The screenshot shows the AWS Management Console for the 'Asia Pacific (Mumbai)' region, user 'Soumili Chatterjee'. The 'EC2' menu is selected, and the 'Instances' page is displayed. A table lists one instance, 'sounew1', with ID 'i-040aa3d4326cba095', in a 'Running' state. The instance type is 't2.micro' and it is in the 'ap-south-1' availability zone. The status check shows 'Initializing'.

The screenshot shows the 'Instance summary' page for 'i-040aa3d4326cba095 (sounew1)'. The instance is in a 'Running' state. Key details include:
 

- Instance ID:** i-040aa3d4326cba095
- Public IPv4 address:** 52.66.241.191
- Private IPv4 addresses:** 172.31.3.188
- Public IPv4 DNS:** ec2-52-66-241-191.ap-south-1.compute.amazonaws.com
- Private IP DNS name (IPv4 only):** ip-172-31-3-188.ap-south-1.compute.internal
- Instance type:** t2.micro
- VPC ID:** vpc-06dd141e6310b8408
- Auto-assigned IP address:** 52.66.241.191 [Public IP]

14. Copy the public IPv4 Address and paste it in a new window followed by a colon and then the port number (4000).

The screenshot shows a web browser window with the address bar containing '52.66.241.191:4000'. The browser tabs include 'Gmail', 'YouTube', 'Maps', 'Canva Resume', and 'Overleaf Resume'. The page content displays 'Hello World'.

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