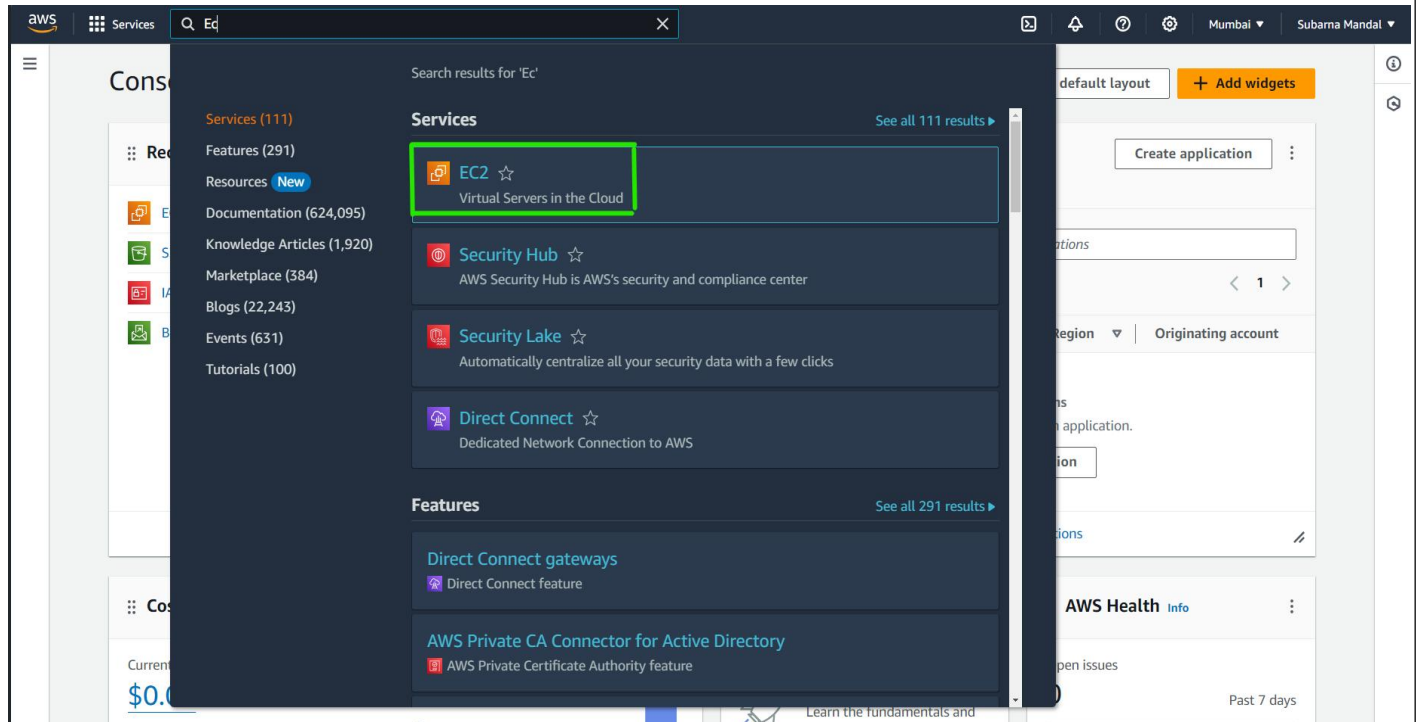


Assignment :- 10

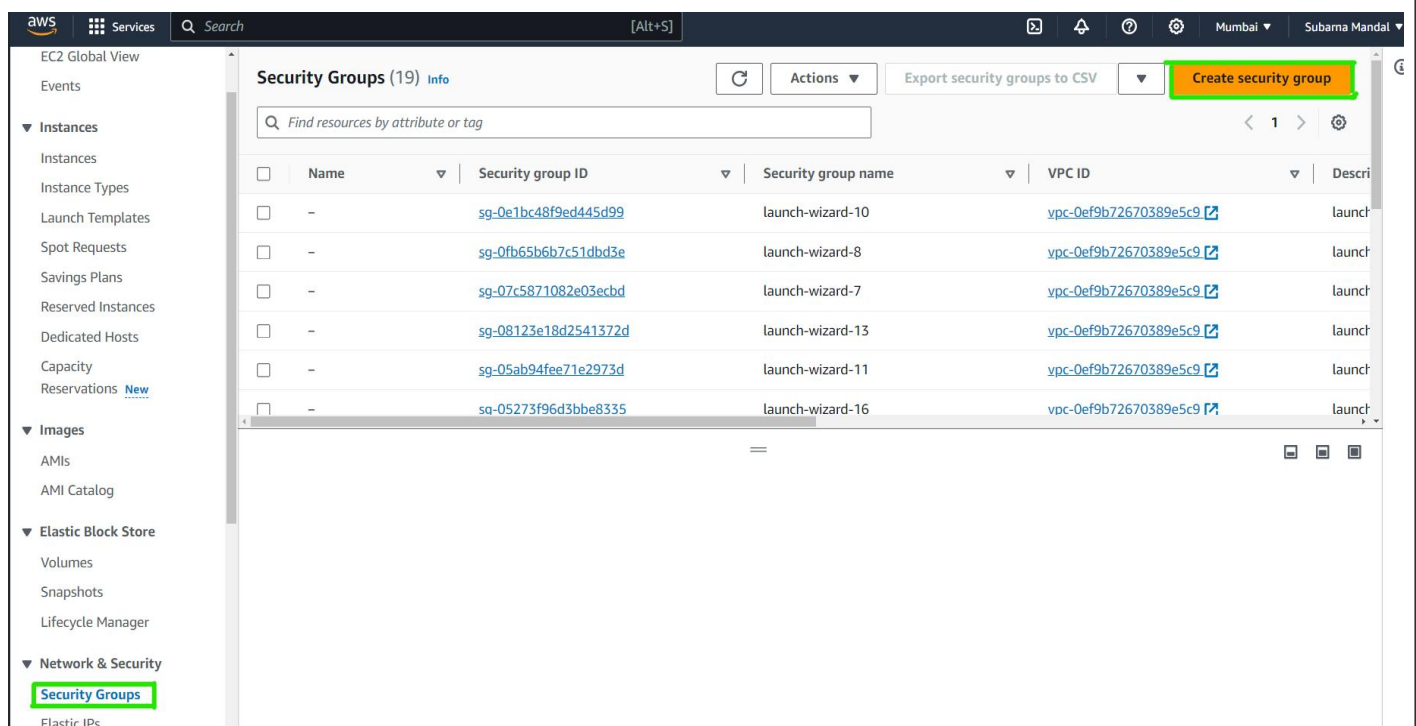
Problem Statement :-

Deploy a project from **GitHub** to **EC2** by creating **new Security Group** and user data.

Step 1 :- Login to the **AWS console**, and search for **EC2**. Open the first EC2 link.



Step 2 :- Go to “**Security Groups**”. Then click on “**Create Security group**” to create custom security group.



Step 3 :- Now provide security group name ,description and Set Inbound rules types **HTTP, HTTPS, SSH** and **Custom TCP Port Range 4000**. Click on the “**Create security group**”.

Basic details

Security group name [Info](#)
MySecurityAWS-10
Name cannot be edited after creation.

Description [Info](#)
For AWS-10

VPC [Info](#)
vpc-0ef9b72670389e5c9

Inbound rules [Info](#)

Type	Protocol	Port range	Source	Description - optional	
Custom TCP	TCP	4000	Anywhere...		Delete
HTTP	TCP	80	Anywhere...		Delete
HTTPS	TCP	443	Anywhere...		Delete
SSH	TCP	22	Anywhere...		Delete

[Add rule](#)

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

[Cancel](#) [Create security group](#)

Step 4 :- Create EC2 ubuntu instance. In **network settings**, select existing security group and choose the group created.

Network settings [Info](#) [Edit](#)

Network [Info](#)
vpc-0ef9b72670389e5c9

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

MySecurityAWS-10 sg-006cd1ca54975633 X
VPC: vpc-0ef9b72670389e5c9

[Compare security group rules](#)

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

Configure storage [Info](#) [Advanced](#)

1x 8 GiB gp3 Root volume (Not encrypted)

[Free tier eligible customers can get up to 30 GB of EBS General Purpose \(SSD\) or Magnetic storage](#) X

[Add new volume](#)

Summary

Number of instances [Info](#)
1

Software image (AMI)
Canonical, Ubuntu, 24.04 LTS, ...read more
ami-0f58b397bc5c1f2e8

Virtual server type (instance type)
t2.micro

Firewall (security group)
MySecurityAWS-10

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

[Cancel](#) [Launch instance](#) [Review commands](#)

Step 5 :- In advanced details,enter user data in the blank space and click “Launch instance”.

The screenshot shows the AWS 'Launch instance' wizard. On the left, the 'User data' field is expanded, showing a script to install Nginx and serve a 'Hello World' page. The script includes commands for updating the system, installing Nginx, enabling it, and setting up a simple web server. On the right, the 'Summary' section shows the instance configuration: 1 instance, Canonical Ubuntu 24.04 LTS AMI, t2.micro instance type, MySecurityAWS-10 security group, and 8 GiB EBS volume. A 'Free tier' notification is visible, stating that the first year includes 750 hours of t2.micro usage. The 'Launch instance' button is orange and prominent.

```
#!/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 https://github.com/SUBARNA-MANDAL/aws10.git
cd aws10
npm install
node index.js
```

Step 6 :- Created instance shown below.

The screenshot displays the 'Instance summary' page for instance i-016ee638f8f17686b. The instance is in a 'Running' state. The 'Details' tab is selected, showing the following information:

- Instance ID:** i-016ee638f8f17686b (aws10)
- Public IPv4 address:** 65.0.73.84 | [open address](#)
- Private IPv4 addresses:** 172.31.8.184
- Public IPv4 DNS:** ec2-65-0-73-84.ap-south-1.compute.amazonaws.com | [open address](#)
- Instance state:** Running
- Private IP DNS name (IPv4 only):** ip-172-31-8-184.ap-south-1.compute.internal
- Instance type:** t2.micro
- VPC ID:** vpc-0ef9b72670389e5c9 | [open address](#)
- Subnet ID:** subnet-033b9ddef7992450c | [open address](#)
- Platform:** Ubuntu (Inferred)
- Platform details:** Linux/UNIX
- Stop protection:** Disabled
- AMI ID:** ami-0f58b397bc5c1f2e8
- AMI name:** ubuntu/images/hvm-ssd-gp3/ubuntu-noble-24.04-amd64-server-20240423
- Launch time:** Tue Apr 30 2024 21:03:35 GMT+0530 (India Standard Time) (3 minutes)
- Monitoring:** disabled
- Termination protection:** Disabled
- AMI location:** [open address](#)

Step 7 :- Now copy the public IPv4 address with custom port number 4000 and run it in any web browser.

The screenshot shows a web browser window with the address bar set to 65.0.73.84:4000. The page content displays 'Hello World'.