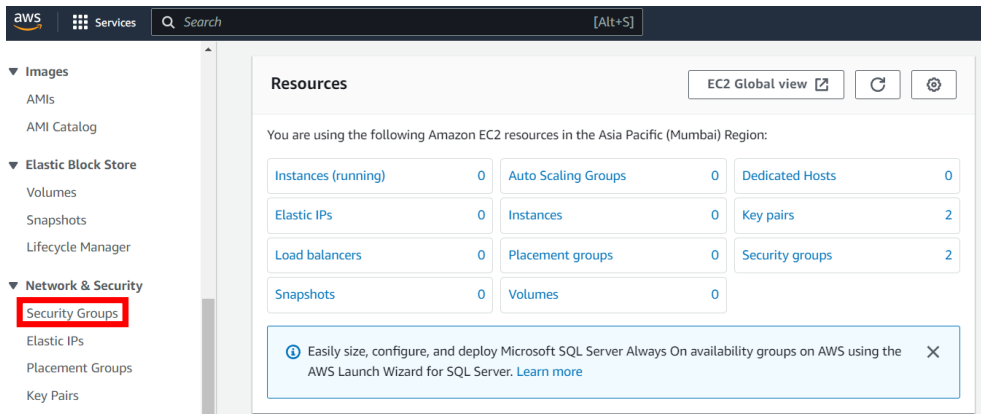


# ASSIGNMENT – 10

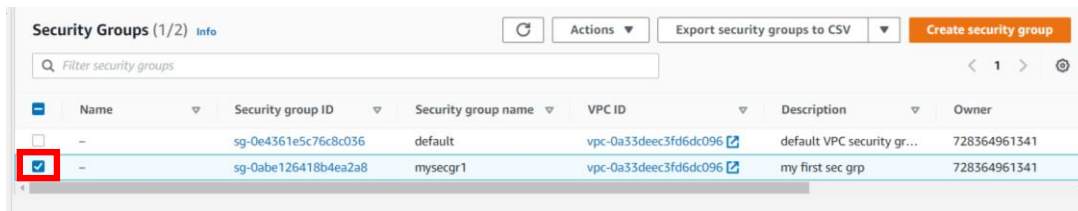
**Problem Statement:** Deploy project from GitHub to EC2 by creating new security group and user data.

## Procedure:

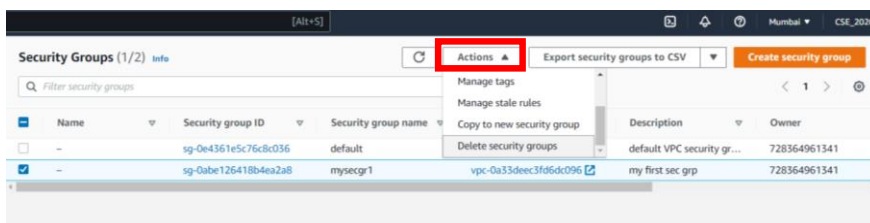
1. Sign in to your AWS account.
2. Go to your EC2 dashboard
3. Scroll down and Click on **Security Groups** option on the left side nav bar under Network & Security option.



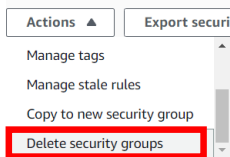
4. Select all the Security Groups other than the one named “default”.



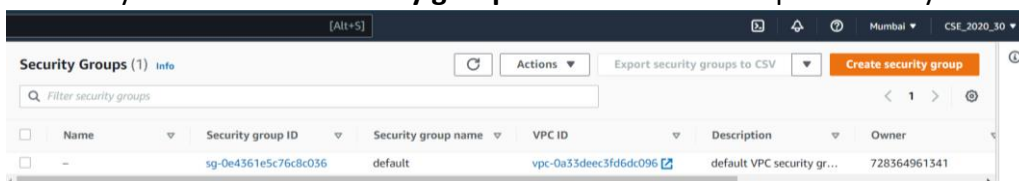
5. Then Click on the **Actions** button.



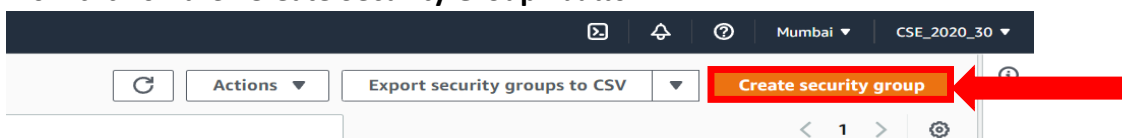
6. Scroll-Down the dropdown list until you find the “delete all security groups” option. Click on it.



7. Now only the “default” security group remains and we keep it that way.



8. Now click on the “Create Security Group” button.



9. Now start by giving a **name** to the **security group** and **giving its description** (anything).

Let the **VPC** remain unchanged.

EC2 > Security Groups > Create security group

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

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

**Basic details**

Security group name [Info](#)

Name cannot be edited after creation.

Description [Info](#)

VPC [Info](#)

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

a) SSH

Type [Info](#) Protocol [Info](#) Port range [Info](#) Source [Info](#) Description - optional [Info](#)

SSH TCP 22 Anywh... 0.0.0.0/0 X Delete

b) HTTP

HTTP TCP 80 Anywh... 0.0.0.0/0 X Delete

c) HTTPS

HTTPS TCP 443 Anywh... 0.0.0.0/0 X Delete

d) Custom TCP

Custom TCP TCP 4000 Anywh... 0.0.0.0/0 X Delete

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 Ass9).

Now the final **Inbound Rules** section should look like this.

Inbound rules [Info](#)

Type <a href="#">Info</a>	Protocol <a href="#">Info</a>	Port range <a href="#">Info</a>	Source <a href="#">Info</a>	Description - optional <a href="#">Info</a>
SSH	TCP	22	Anywh... 0.0.0.0/0 X	
HTTP	TCP	80	Anywh... 0.0.0.0/0 X	
HTTPS	TCP	443	Anywh... 0.0.0.0/0 X	
Custom TCP	TCP	4000	Anywh... 0.0.0.0/0 X	

Add rule

11. Next **outbound rules** and all other sections remain unchanged. Now Click on the **create security group** button.

Outbound rules [Info](#)

Type <a href="#">Info</a>	Protocol <a href="#">Info</a>	Port range <a href="#">Info</a>	Destination <a href="#">Info</a>	Description - optional <a href="#">Info</a>
All traffic	All	All	Custom 0.0.0.0/0 X	

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

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

Security Groups (2) <a href="#">Info</a>			
<input type="text" value="Filter security groups"/>			
<input type="checkbox"/>	Name	Security group ID	Security group name
<input type="checkbox"/>	-	sg-0493398d43b761e55	mysec1
<input type="checkbox"/>	-	sg-0e4361e5c76c8c036	default

Security group name  
mysec1

Security group ID  
sg-0493398d43b761e55

Description  
mysec1

VPC ID  
vpc-0a33deec3fd6dc096

Owner  
728364961341

Inbound rules count  
4 Permission entries

Outbound rules count  
1 Permission entry

Inbound rules

Outbound rules

Tags

You can now check network connectivity with Reachability Analyzer [Run Reachability Analyzer](#)

Inbound rules (4)

☐

Name

Security group rule...

IP version

Type

Protocol

Port range

☐

-

sg-0b77b32c36bf02194

IPv4

HTTPS

TCP

443

☐

-

sg-02f9143435809d0...

IPv4

SSH

TCP

22

☐

-

sg-08f9ba0e0aecca64

IPv4

HTTP

TCP

80

☐

-

sg-0d92a3e25bf3add37

IPv4

Custom TCP

TCP

4000

After clicking we can view the inbound rules that we added during its creation.

13. Now we go to the instances section from the left side nav bar.

14. Now we **Create a new EC2 instance**. Click on the **Launch Instance** button.

Instances [Info](#)

Connect

Instance state

Actions

Launch instances

Name

Instance ID

Instance state

Instance type

Status check

Alarm status

Availability Zone

Public IPv4 DNS

No instances

You do not have any instances in this region

Now,

a) Give the name

Launch an instance [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags [Info](#)

Name

debserver1

Add additional tags

b) Select Ubuntu as OS.

Application and OS Images (Amazon Machine Image) [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

Quick Start

Amazon Linux

macOS

Ubuntu

Windows

Red Hat

S

aws

Mac

ubuntu

Microsoft

Red Hat

Browse more AMIs

c) Select a keypair or generate a new one if none is available.

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

debkey2

Create new key pair

d) Then under Network settings select the Select Existing Security Group option.

Network settings [Info](#)

Edit

Network [Info](#)

vpc-0a33deec3fd6dc096

Subnet [Info](#)

No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)

Enable

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

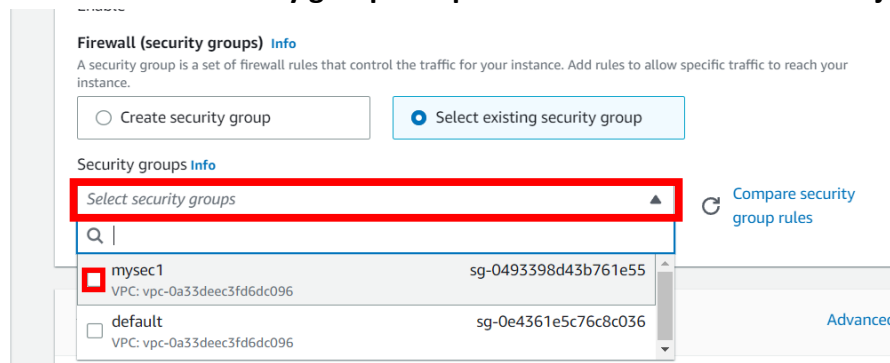
Select existing security group

Security groups [Info](#)

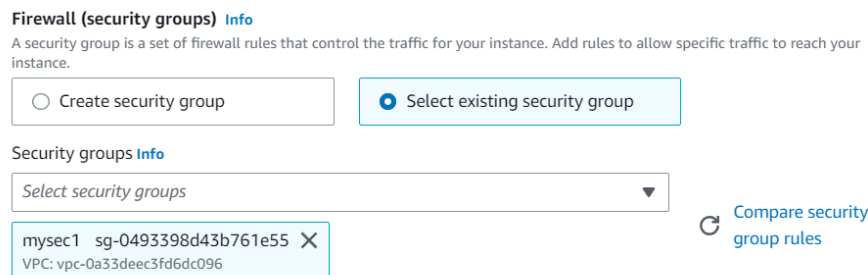
Select security groups

Compare security group rules

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



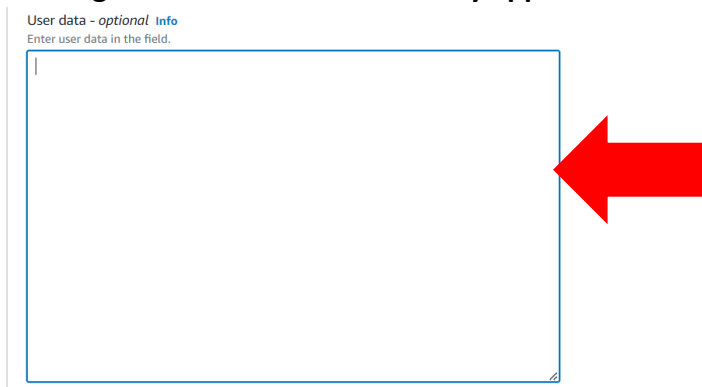
It should look like this.....



f) Now scroll down and click on the Advanced Details option.



g) Now again scroll-down to the newly appeared sub-sections until you find User Data section.



h) Write the following commands in the given box. Remember this user data is given to execute the given commands once the server starts. So essentially, we can provide all commands that we entered in our Assignment 9 previously and execute them without connecting to our server itself!! They will be executed sequentially.

```
#!/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
```

Now, here is a caveat. **We have 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.

i) Now we click on the launch instance button.

User data - optional [Info](#)  
Enter user data in the field.

```
#!/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
```

☐ User data has already been base64 encoded

Software Image (AMI)  
Canonical, Ubuntu, 22.04 LTS, ...[read more](#)  
ami-02eb7a4783e7e9317

Virtual server type (instance type)  
t2.micro

Firewall (security group)  
mysec1

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

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

15. Now we Click on the 'Instance Id' link of our newly created server in our Instances list.

Instances (1) <a href="#">Info</a>							
<input type="text" value="Find instance by attribute or tag (case-sensitive)"/>							
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input type="checkbox"/>	debserver1	<b>i-0a6ab24417f81fffb</b>	Running	t2.micro	Initializing	No alarms +	ap-south-1a

16. Now click on the connect button

Instance summary for i-0a6ab24417f81fffb (debserver1) [Info](#)

Connect

Instance state

Actions

Instance ID  
i-0a6ab24417f81fffb (debserver1)

Public IPv4 address  
3.110.134.34 | [open address](#)

Private IPv4 addresses  
172.31.41.246

IPv6 address  
-

Instance state  
Running

Public IPv4 DNS  
ec2-3-110-134-34.ap-south-1.compute.amazonaws.com | [open address](#)

17. Again, click on the connect button

Connect to instance [Info](#)

Connect to your instance i-0a6ab24417f81fffb (debserver1) using any of these options

EC2 Instance Connect

Session Manager

SSH client

EC2 serial console

Instance ID  
i-0a6ab24417f81fffb (debserver1)

Public IP address  
3.110.134.34

User name  
ubuntu

Note: In most cases, the default user name, ubuntu, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

Cancel **Connect**

18. After this, a new Tab will open with a Bash Terminal that is of our remote EC2 server!

Here, we can type all our required commands that we used to type in a similar terminal by connecting to our remote server through our Bitwise SSH client software in our previous assignments.

```
aws Services Search [Alt+S] Mumbai CSE_2020_30
Receive updates to over 25,000 software packages with your
Ubuntu Pro subscription. Free for personal use.

https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

33 updates can be applied immediately.
18 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

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.

ubuntu@ip-172-31-41-246:~$
```

19. Now type the following commands in the terminal :-

→ **git clone** <https://github.com/>..... //Your GitHub Repository URL

(Give your Username of GitHub when asked.)

(Give your account Token when your Password is asked.)

```
ubuntu@ip-172-31-41-246:~$ git clone https://github.com/DebrupPramanik/myRepoV1.git
Cloning into 'myRepoV1'...
Username for 'https://github.com': DebrupPramanik
Password for 'https://DebrupPramanik@github.com':
remote: Enumerating objects: 15, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (14/14), done.
remote: Total 15 (delta 6), reused 4 (delta 0), pack-reused 0
Receiving objects: 100% (15/15), done.
Resolving deltas: 100% (6/6), done.
```

→ **cd YourRepositoryname/**

```
ubuntu@ip-172-31-41-246:~$ cd myRepoV1/
ubuntu@ip-172-31-41-246:~/myRepoV1$
```

→ **npm install**

```
ubuntu@ip-172-31-41-246:~/myRepoV1$ npm install
npm WARN deprecated uuid@3.4.0: Please upgrade to version 7 or higher. Or see https://v8.dev/blog/math-random for details.

added 258 packages, and audited 259 packages in 15s

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

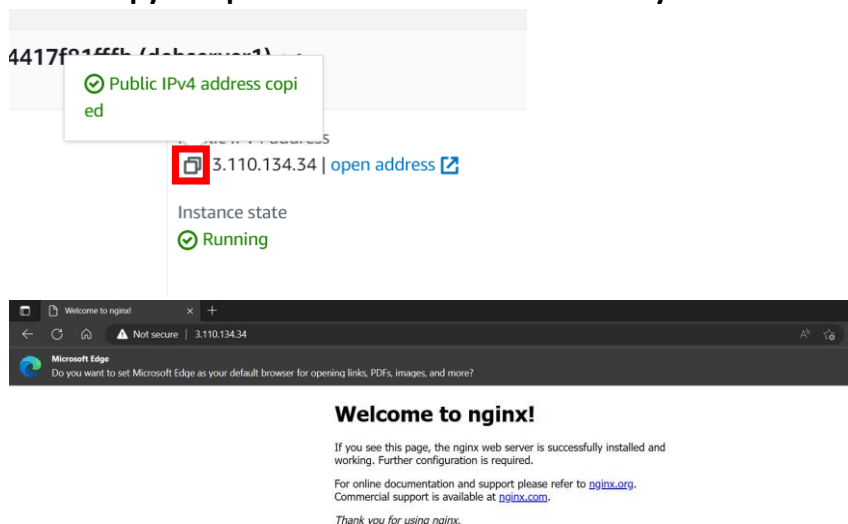
found 0 vulnerabilities

npm notice
npm notice New minor version of npm available! 9.5.1 -> 9.6.5
npm notice Changelog: https://github.com/npm/cli/releases/tag/v9.6.5
npm notice Run npm install -g npm@9.6.5 to update!
npm notice
```

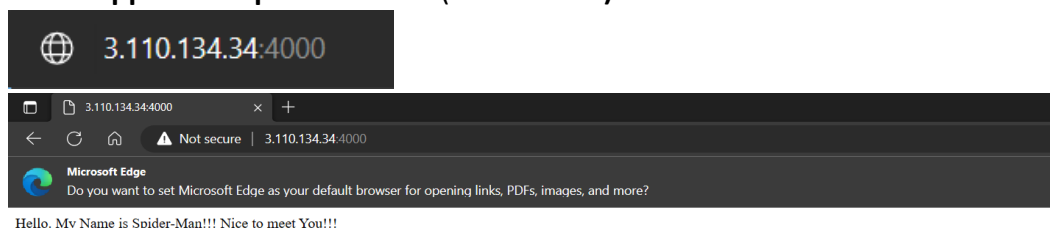
→ **node index.js**

```
ubuntu@ip-172-31-41-246:~/myRepoV1$ node index.js
Started server
```

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



21. 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.**