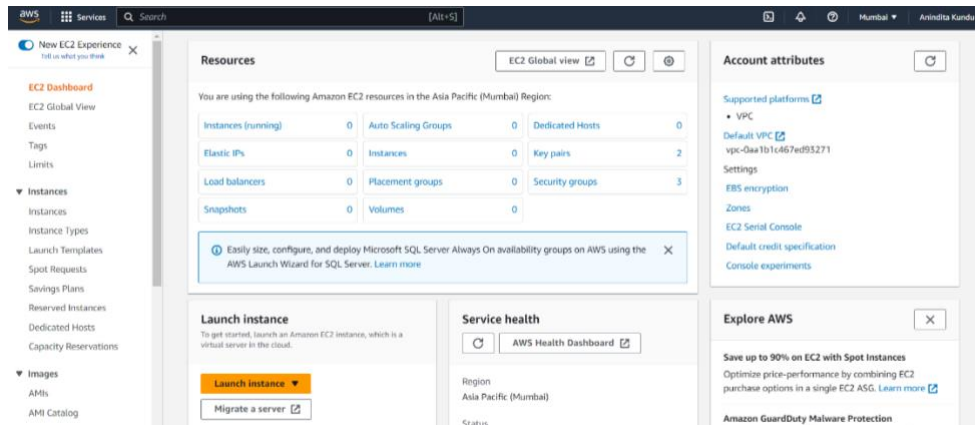


ASSIGNMENT – 10

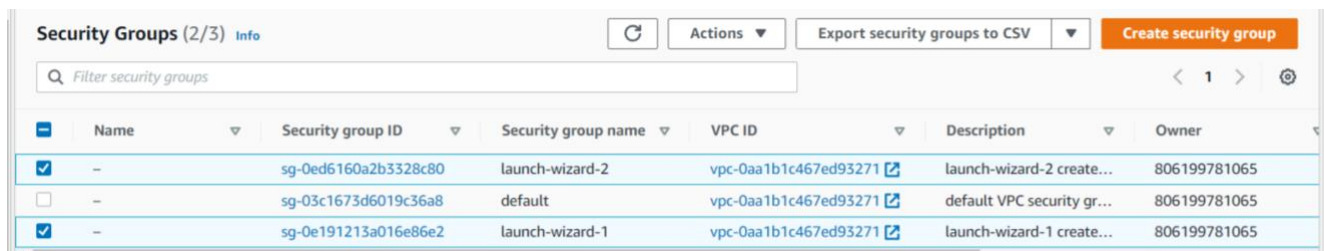
Problem Statement: Deploy project from GitHub to EC2 by creating new security group and user data.

Procedure:

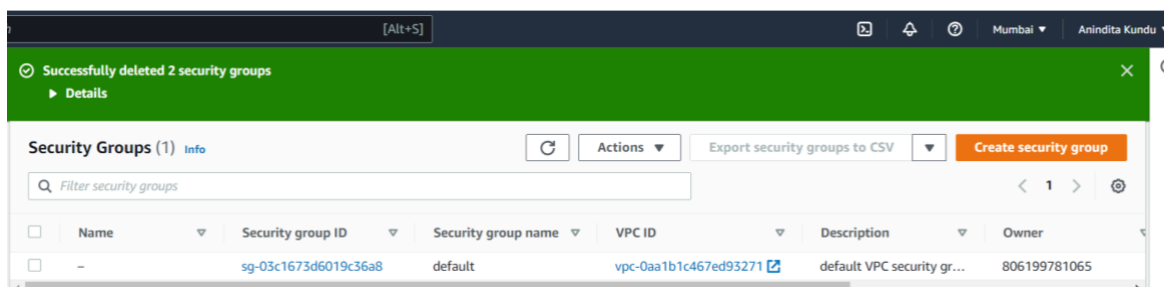
Step 1: Sign in to your AWS account. Go to your EC2 dashboard. Scroll down and Click on Security Groups option on the left side nav bar under Network & Security option.



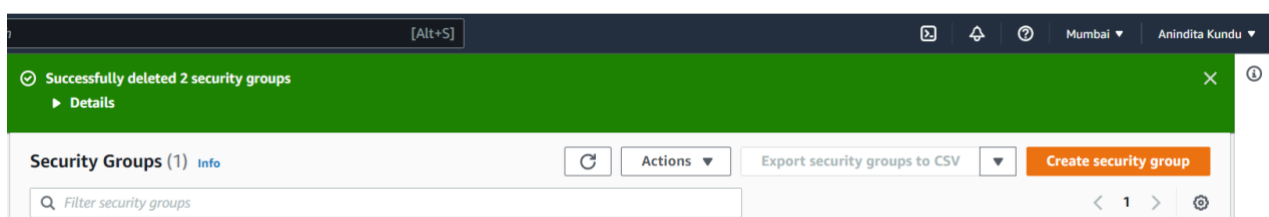
Step 2: Select all the Security Groups other than the one named “default”.



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.

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](#)
A1
Name cannot be edited after creation.

Description [Info](#)
A1

VPC [Info](#)
vpc-0aa1b1c467ed93271

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).

Inbound rules [Info](#)

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

Add rule

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

Outbound rules [Info](#)

Type Info	Protocol Info	Port range Info	Destination Info	Description - optional Info
All traffic	All	All	Custom 0.0.0.0/0	

Add rule

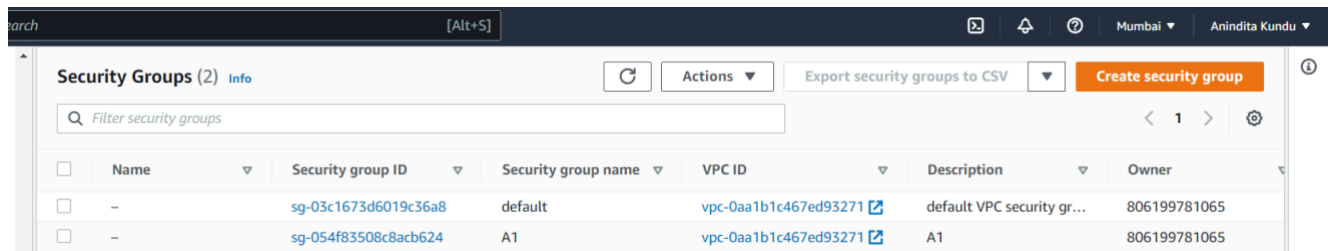
Tags - optional
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

No tags associated with the resource.

Add new tag
You can add up to 50 more tags

Cancel Create security group

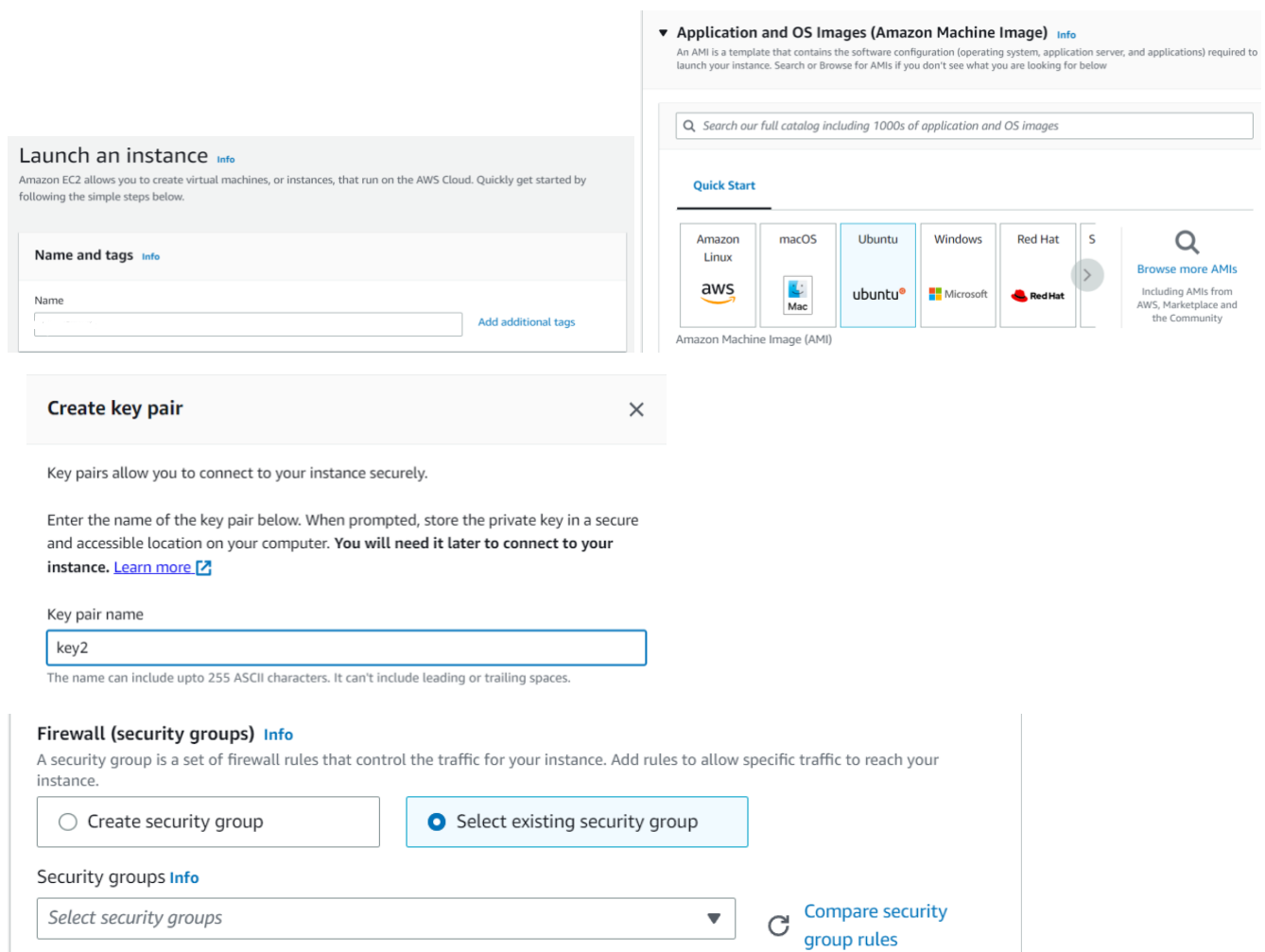
Step 8: Now go back to the security groups list and click on the security group ID of the newly created Security Group. After clicking we can view the inbound rules that we added during its creation.



Security Groups (2) Info						
Filter security groups						
<input type="checkbox"/>	Name	Security group ID	Security group name	VPC ID	Description	Owner
<input type="checkbox"/>	-	sg-03c1673d6019c36a8	default	vpc-0aa1b1c467ed93271	default VPC security gr...	806199781065
<input type="checkbox"/>	-	sg-054f83508c8acb624	A1	vpc-0aa1b1c467ed93271	A1	806199781065

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.



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 [Add additional tags](#)

Create key pair X

Key pairs allow you to connect to your instance securely.

Enter the name of the key pair below. When prompted, store the private key in a secure and accessible location on your computer. **You will need it later to connect to your instance.** [Learn more](#)

Key pair name

The name can include upto 255 ASCII characters. It can't include leading or trailing spaces.

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

[Compare security group rules](#)

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

Amazon Linux macOS Ubuntu Windows Red Hat S

Amazon Machine Image (AMI)

[Browse more AMIs](#)

Including AMIs from AWS, Marketplace and the Community

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.

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
git clone https://github.com/Anindita-11/MyRepo2.git
cd MyRepo2/
npm install
node index.js
```

☐ User data has already been base64 encoded

ami-02eb7a4783e7e9317

Virtual server type (instance type)
t2.micro

Firewall (security group)
A1

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 GiB of EBS storage.

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

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

Instance summary for i-0cbbcad058fbafd9b (ec2anindita) [Info](#)

Updated less than a minute ago

[Refresh](#) [Connect](#) [Instance state ▼](#) [Actions ▼](#)

Instance ID i-0cbbcad058fbafd9b	Public IPv4 address [redacted]	Private IPv4 addresses [redacted]
IPv6 address -	Instance state Running	Public IPv4 DNS [redacted]

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 Bitwise 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
see 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.

Welcome to nginx!




If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

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

← → ↻ ⚠ Not secure | 13.233.225.86:4000

 Gmail  YouTube  Maps

Hello, Anindita Here

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