

# Assignment 10

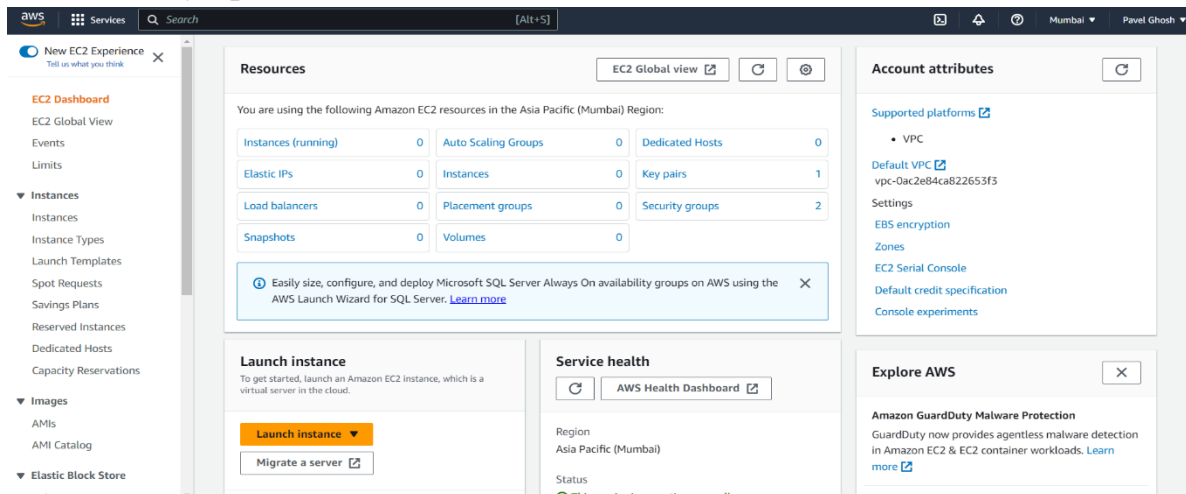
**Problem Statement:** Deploy project from GitHub to EC2 by creating new security group and user data.

**Procedure:**

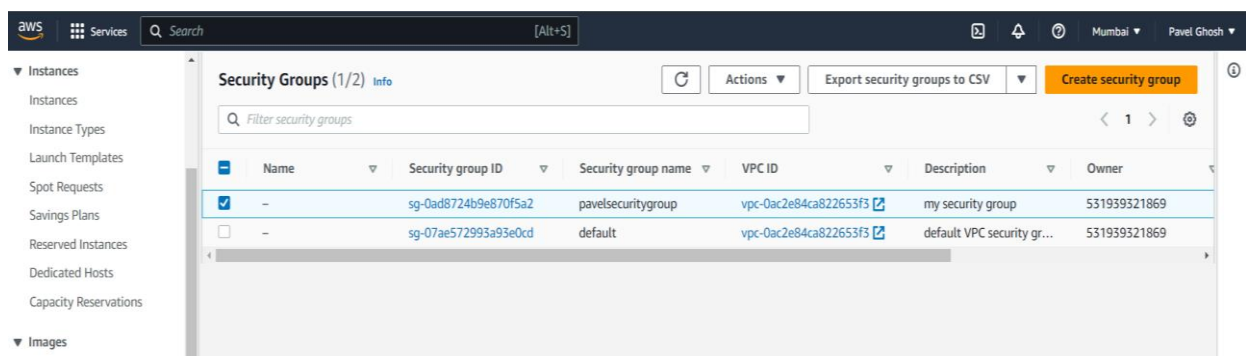
**Step 1:** Sign in to your AWS account.

**Step 2:** Go to your EC2 dashboard

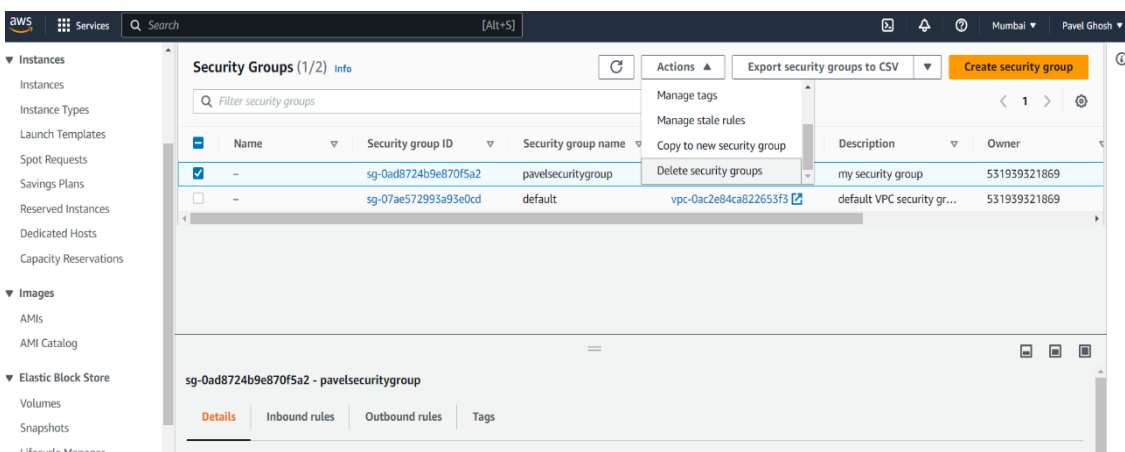
**Step 3:** Scroll down and Click on Security Groups option on the left side nav bar under Network & Security option.



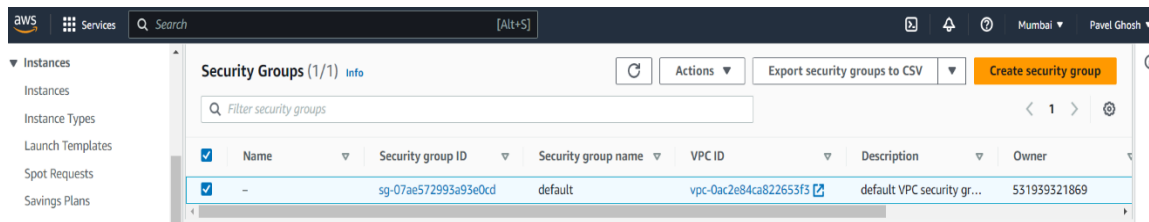
**Step 4:** Select all the Security Groups other than the one named “default”.



**Step 5:** Then Click on the Actions button. Scroll-Down the dropdown list until you find the “delete all security groups” option. Click on it.



**Step 6:** Now only the “default” security group remains and we keep it that way.



**Step 7:** Now click on the “Create Security Group” button.

**Step 8:** Now start by giving a name to the security group and giving its description (anything).

A screenshot of the 'Create security group' form in the AWS Management Console. The form is titled 'Create security group' and includes a sub-section 'Basic details'. It contains three input fields: 'Security group name' with the value 'myserver7', 'Description' with the value 'myserver7', and 'VPC' with the value 'vpc-0ac2e84ca822653f3'. There are also 'Info' links for each field and a note stating 'Name cannot be edited after creation.'

**Step 9:** Next, we will add Inbound Rules. Start adding by clicking the Add rule button. These include:

A screenshot of the 'Inbound rules' section in the AWS Management Console. It displays a table with four rules. Each rule row includes a 'Type' dropdown, a 'Protocol' dropdown, a 'Port range' input, a 'Source' dropdown with a search box, and a 'Description - optional' input. The rules are: SSH (TCP, 22, Anywh...), HTTP (TCP, 80, Anywh...), HTTPS (TCP, 443, Anywh...), and Custom TCP (TCP, 4000, Anywh...). Each rule has a 'Delete' button. At the bottom left, there is an 'Add rule' button.

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).

**Step 10:** 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 ▼

**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.

You can add up to 50 more tags

**Step 11:** Now go back to the security groups list and click on the security group ID of the newly created Security Group.

EC2 > Security Groups > sg-0f75e72c39f5ea22c - myserver7

**sg-0f75e72c39f5ea22c - myserver7** [Actions](#) ▼

**Details**

Security group name myserver7	Security group ID sg-0f75e72c39f5ea22c	Description myserver7	VPC ID vpc-0ac2e84ca822653f3
Owner 531939321869	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](#)

**Inbound rules (4)**

<input type="checkbox"/>	Name	Security group rule...	IP version	Type	Protocol	Port range
<input type="checkbox"/>	-	sgr-0e463e727d2276...	IPv4	Custom TCP	TCP	4000
<input type="checkbox"/>	-	sgr-02ee3b63d159ba2...	IPv4	SSH	TCP	22
<input type="checkbox"/>	-	sgr-0dd7f7d6a3128aabd	IPv4	HTTPS	TCP	443
<input type="checkbox"/>	-	sgr-0d70feb21b51849...	IPv4	HTTP	TCP	80

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

**Step 12:** Now we go to the instances section from the left side nav bar.

**Step 13:** Now we Create a new EC2 instance. Click on the Launch Instance button.

**Instances** [Info](#)  [Instance state](#) ▼ [Actions](#) ▼

<input type="checkbox"/>	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								
<input type="button" value="Launch instances"/>								

Now,

- Give the name
- Select Ubuntu as OS

**Name and tags** [Info](#)

Name  
pavelserv7 [Add additional tags](#)

▼ **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

Search your full catalog including 1000s of application and OS images

**Quick Start**

Amazon Linux macOS Ubuntu Windows Red Hat S [Browse more AMIs](#)  
Including AMIs from AWS, Marketplace and the Community

**Summary**

Number of instances [Info](#)  
1

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

Virtual server type (instance type)  
t2.micro

Firewall (security group)  
New security group

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

**Free tier:** In your first year includes 750 hours of t2.micro (or t3.micro in the

Cancel [Launch instance](#)

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*  
pavel7 [Create new key pair](#)

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

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

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

Network [Info](#)  
vpc-0ac2e84ca822653f3

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

myserver7 sg-0f75e72c39f5ea22c [X](#)  
VPC: vpc-0ac2e84ca822653f3 [Compare security group rules](#)

It should look like this...

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

► **Advanced details** [Info](#)

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

User data - optional [Info](#)

Enter user data in the field.

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.

- i) **#!/bin/bash**
- j) **apt-get update**
- k) **apt-get install -y nginx**
- l) **systemctl start nginx**
- m) **systemctl enable nginx**
- n) **apt-get install -y git**
- o) **curl -sL https://deb.nodesource.com/setup\_18.x | sudo -E bash -**
- p) **apt-get install -y nodejs**
- q) **git clone https://github.com/itzFelu/collegeRepo.git**
- r) **cd collegeRepo**
- s) **npm install**  
**node index.js**

Now, here is a caveat. 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.

t) Now we click on the launch instance button.

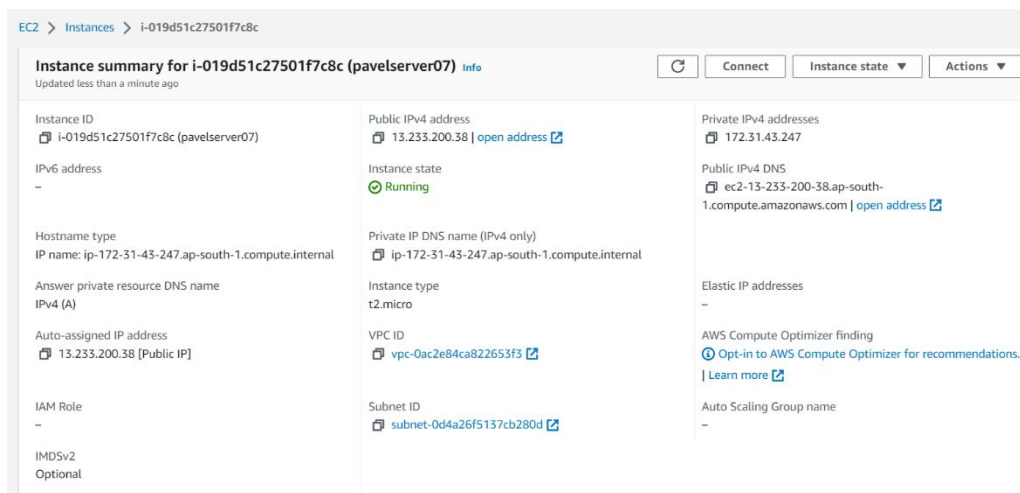
User data - optional [Info](#)

Enter user data in the field.

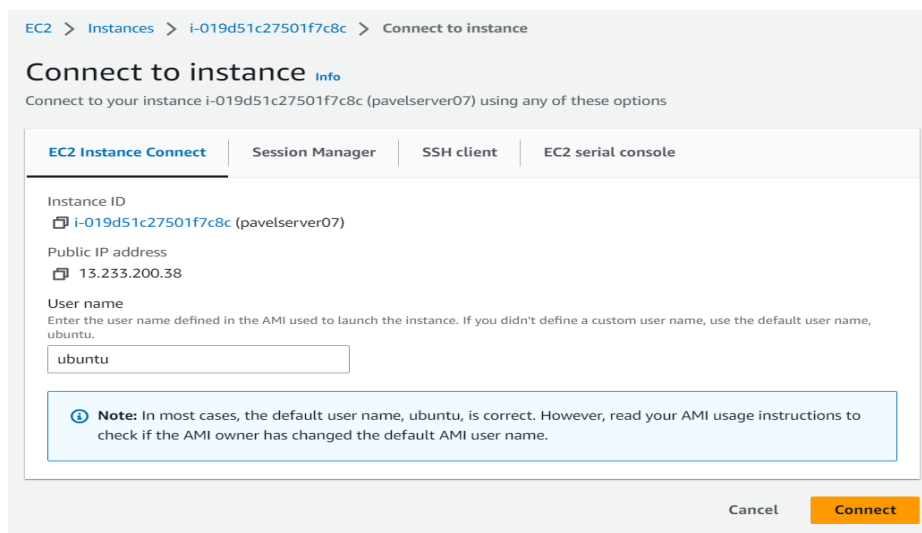
```
#!/bin/bash
apt-get update
apt-get install -y nginx
systemctl start nginx
systemctl enable nginx
apt-get install -y git
curl -sL https://deb.nodesource.com/setup_18.x | sudo -E bash -
apt-get install -y nodejs
git clone https://github.com/itzFelu/collegeRepo.git
cd collegeRepo
npm install
node index.js
```

**Step 14:** Now we Click on the ‘Instance Id’ link of our newly created server in our Instances list.

**Step 15:** Now click on the connect button.



**Step 16:** Again, click on the connect button.



**Step 17:** After this anew 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.

```
Usage of /: 26.2% of 7.57GB    Users logged in: 0
Memory usage: 29%           IPv4 address for eth0: 172.31.43.247
Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.

48 updates can be applied immediately.
23 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-172-31-43-247:~$
```

**Step 18:** 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.

- `cd Your Repository name/`

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

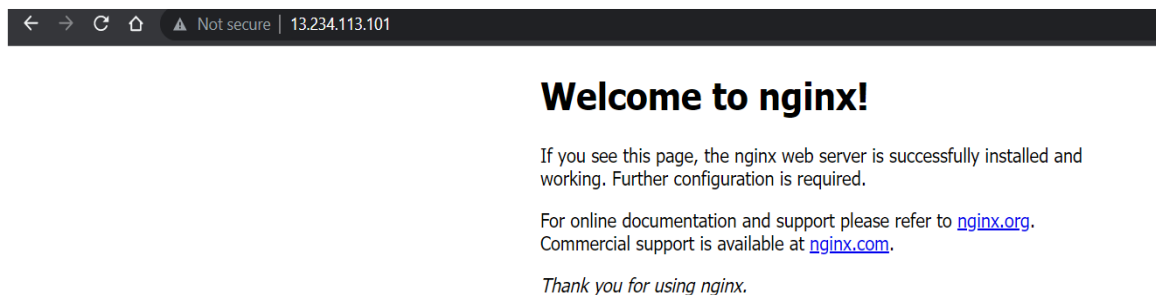
- `npm install`

```
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
```

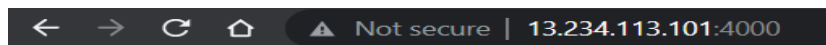
- `node index.js`

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

**Step 19:** Now copy and paste the Public IPv4 address of your EC2 instance in another browser.



**Step 20:** Now append the port no. 4000 (for our case) to the IP address in the browser with a “:” sign.



# Hello

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