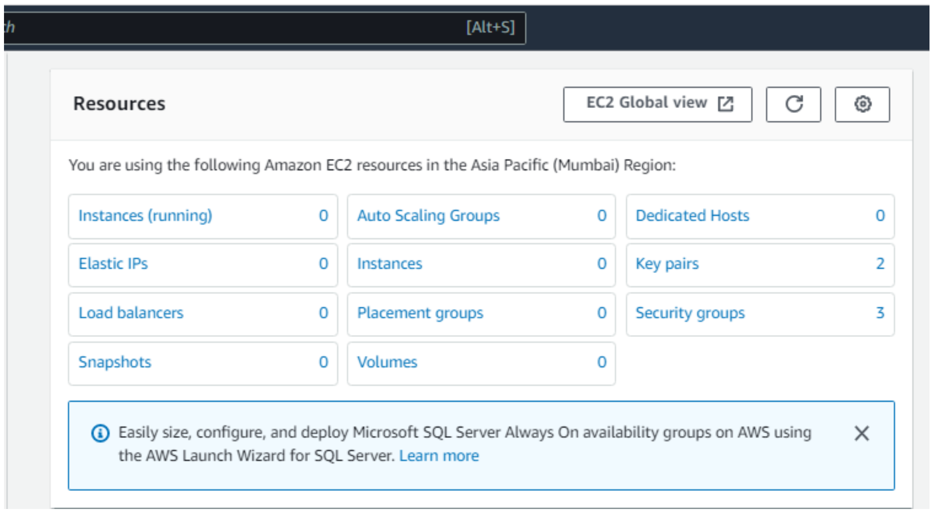
ASSIGNMENT-10

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

Step 1: Sign in to your AWS Account as Root User. Go to EC2 dashboard . And click on Security groups.



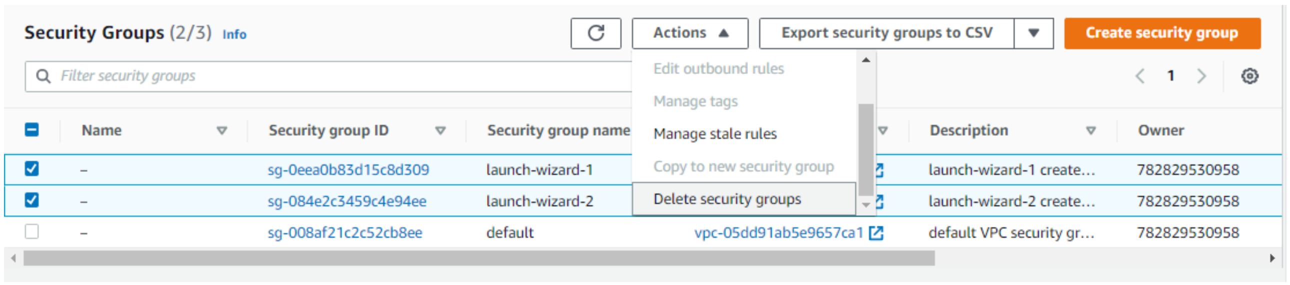
Resources

You are using the following Amazon EC2 resources in the Asia Pacific (Mumbai) Region:

Instances (running)	0	Auto Scaling Groups	0	Dedicated Hosts	0
Elastic IPs	0	Instances	0	Key pairs	2
Load balancers	0	Placement groups	0	Security groups	3
Snapshots	0	Volumes	0		

Easily size, configure, and deploy Microsoft SQL Server Always On availability groups on AWS using the AWS Launch Wizard for SQL Server. Learn more

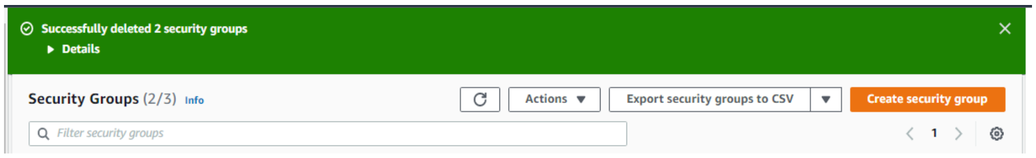
Step 2: Remove all security groups except the default one.



Security Groups (2/3) Info

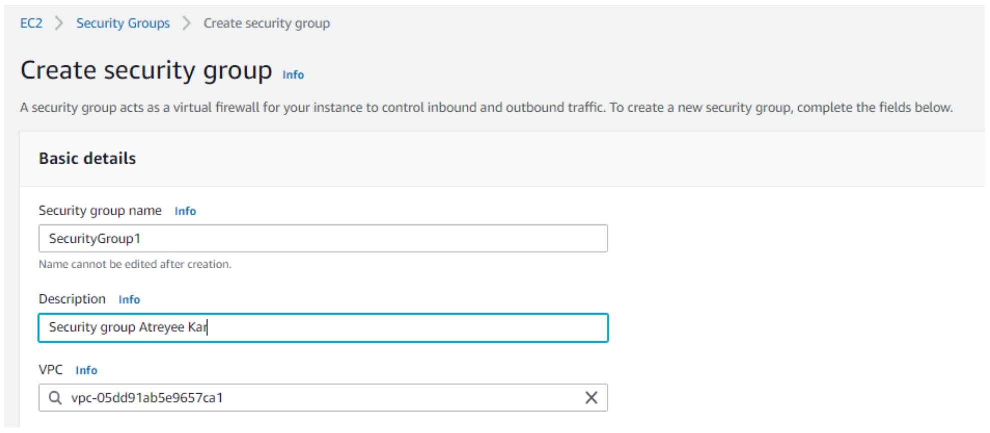
	Name	Security group ID	Security group name	VPC ID	Description	Owner
✓	–	sg-0eea0b83d15c8d309	launch-wizard-1		launch-wizard-1 create...	782829530958
✓	–	sg-084e2c3459c4e94ee	launch-wizard-2		launch-wizard-2 create...	782829530958
	–	sg-008af21c2c52cb8ee	default	vpc-05dd91ab5e9657ca1	default VPC security gr...	782829530958

Step 3: All security groups except the default one is deleted. Now go to "Create security group".



Successfully deleted 2 security groups

Step 4: Now give Security group name and Description. Keep the VPC 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.

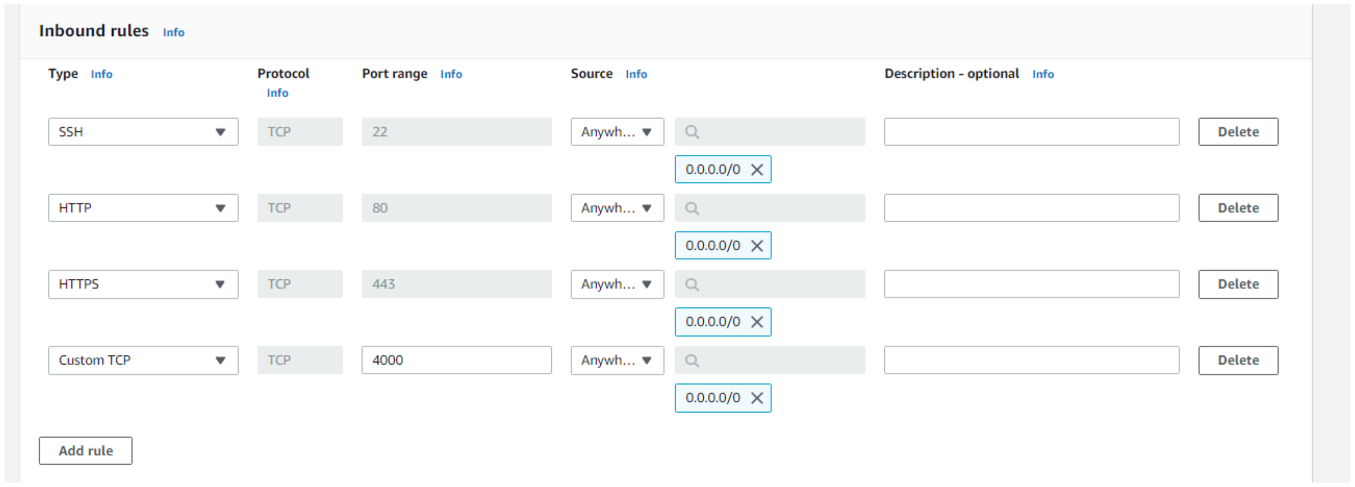
Basic details

Security group name Info
SecurityGroup1
Name cannot be edited after creation.

Description Info
Security group Atreyee Kar

VPC Info
vpc-05dd91ab5e9657ca1

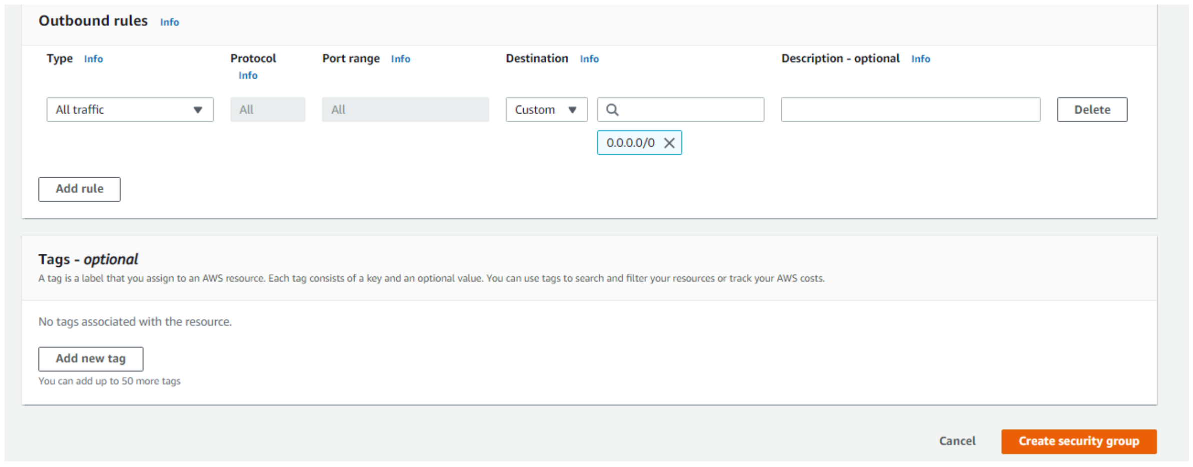
Step 5: Now go to “Inbound rules” and “Add rules” as shown below:


Inbound rules Info

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

Add rule

Step 6: Keep other options unchanged an then “Create security group”.


Outbound rules Info

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

Add rule

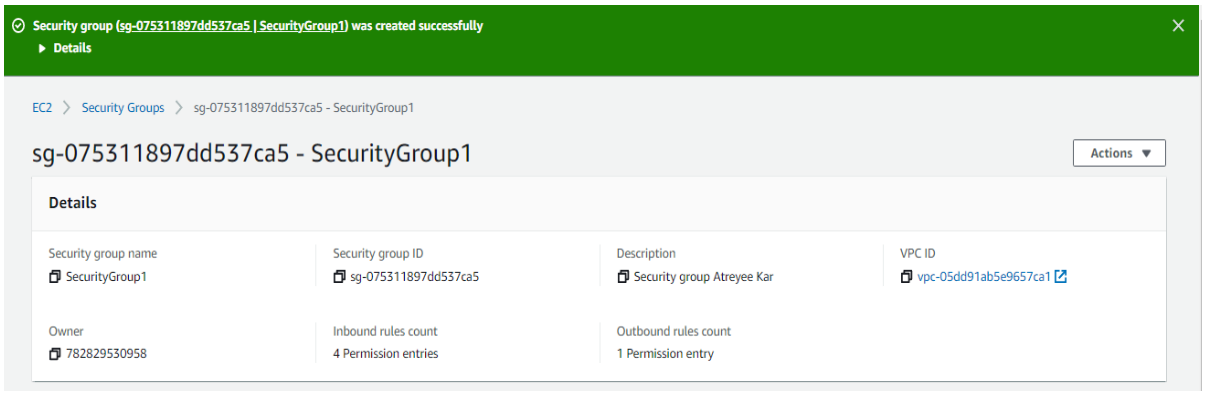
Tags - *optional*
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

No tags associated with the resource.

Add new tag
You can add up to 50 more tags

Cancel Create security group

Step 7: Our Security group is created.



Security group (sg-075311897dd537ca5 | SecurityGroup1) was created successfully

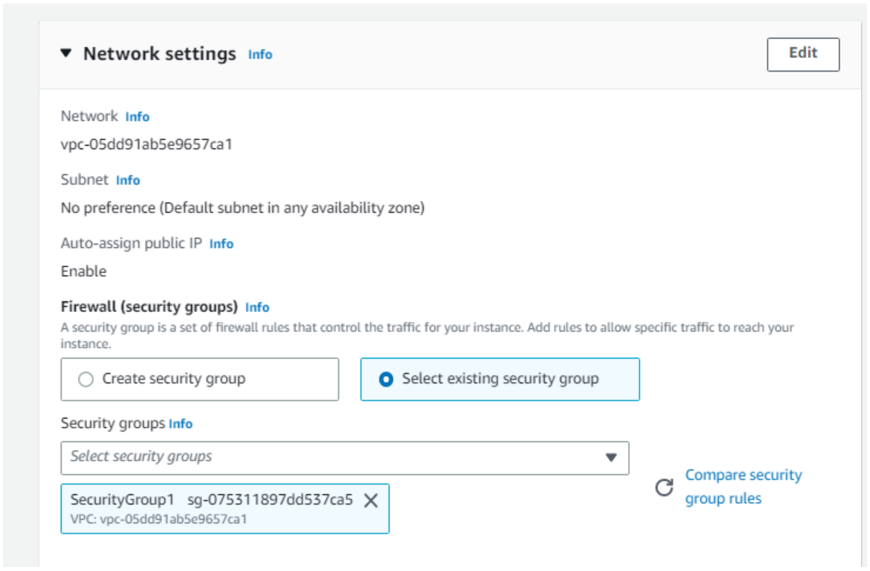
EC2 > Security Groups > sg-075311897dd537ca5 - SecurityGroup1

sg-075311897dd537ca5 - SecurityGroup1

Details

Security group name	Security group ID	Description	VPC ID
SecurityGroup1	sg-075311897dd537ca5	Security group Atreyee Kar	vpc-05dd91ab5e9657ca1
Owner	**Inbound rules count**	**Outbound rules count**	
782829530958	4 Permission entries	1 Permission entry	

Step 8: Create an EC2 instance. Give name. Select OS and keypair. Then under "Network settings", select "Select existing security group", go to Security groups and select the group you have created.



Network settings

Network: vpc-05dd91ab5e9657ca1

Subnet: No preference (Default subnet in any availability zone)

Auto-assign public IP: Enable

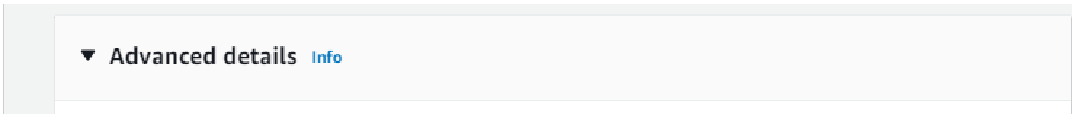
Firewall (security groups)

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]

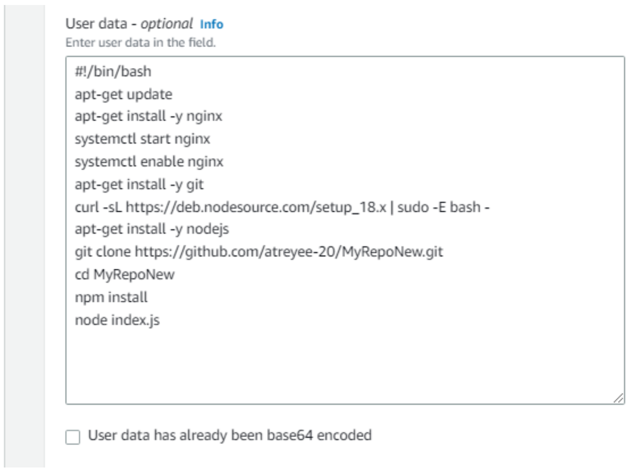
Security groups: SecurityGroup1 sg-075311897dd537ca5 VPC: vpc-05dd91ab5e9657ca1

Step 9: Now scroll down and click on "Advanced details".



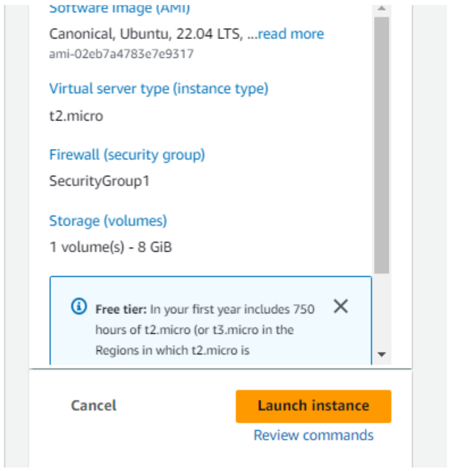
Step 10: Scroll down and write the following commands in “User Data” section as shown:
Commands:

- **#!/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 <The URL of the project>**
- **cd <Repository name of your project>**
- **npm install**
- **node <.js file name>**


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/atreyee-20/MyRepoNew.git
cd MyRepoNew
npm install
node index.js
```

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

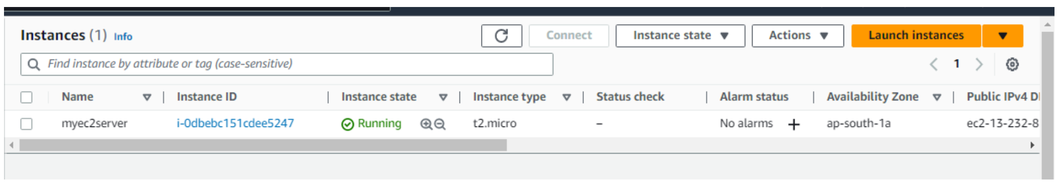
Firewall (security group)
SecurityGroup1

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

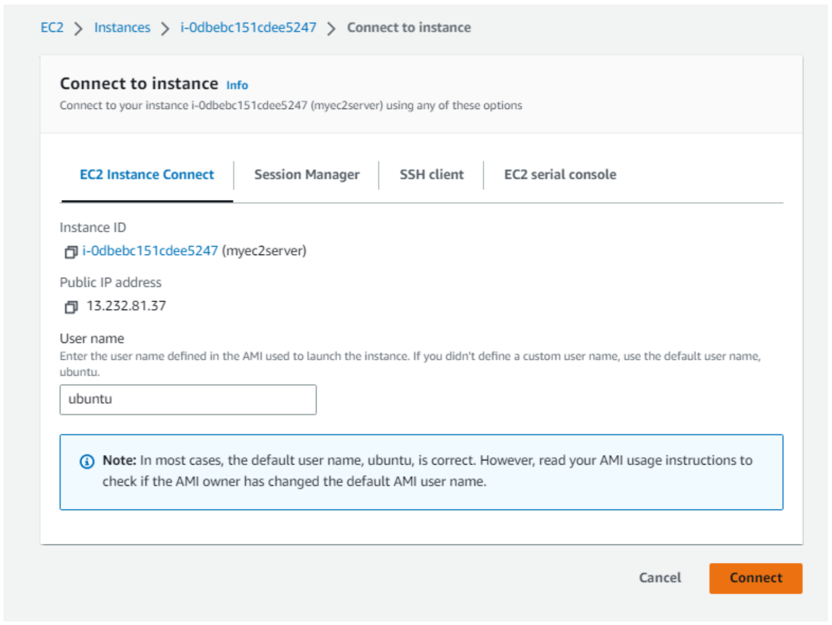
Then click on “Launch instance”.

Step 11: New instance is created. Select the instance and click on the connect button.



Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 D
myec2server	i-0dbebc151cdee5247	Running	t2.micro	–	No alarms	ap-south-1a	ec2-13-232-8

Step 12: Again click on "Connect".


EC2 > Instances > i-0dbebc151cdee5247 > Connect to instance

Connect to instance Info

Connect to your instance i-0dbebc151cdee5247 (myec2server) using any of these options

Instance ID

i-0dbebc151cdee5247 (myec2server)

Public IP address

13.232.81.37

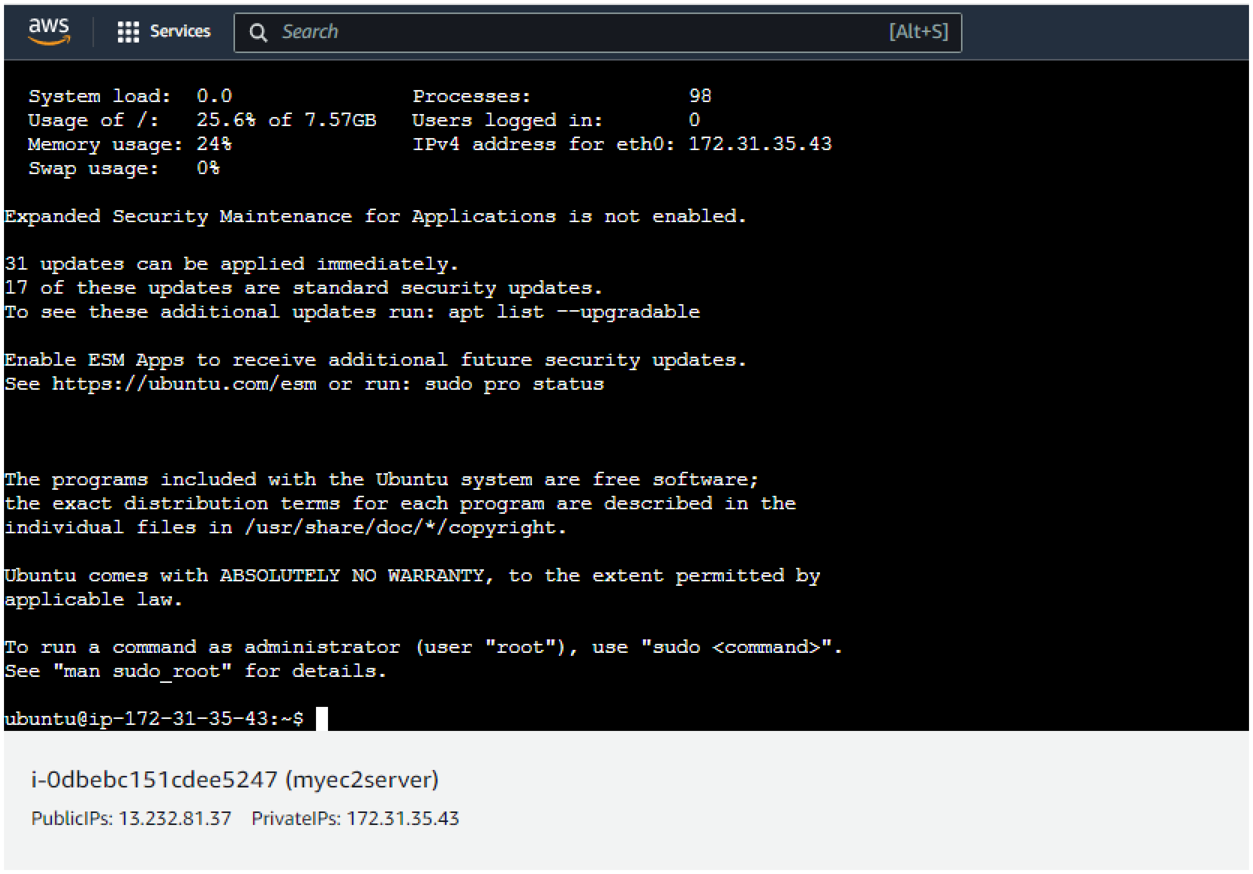
User name

Enter the user name defined in the AMI used to launch the instance. If you didn't define a custom user name, use the default user name, ubuntu.

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.

Step 13: A new tab will open with a Bash Terminal. It is our remote EC2 server.

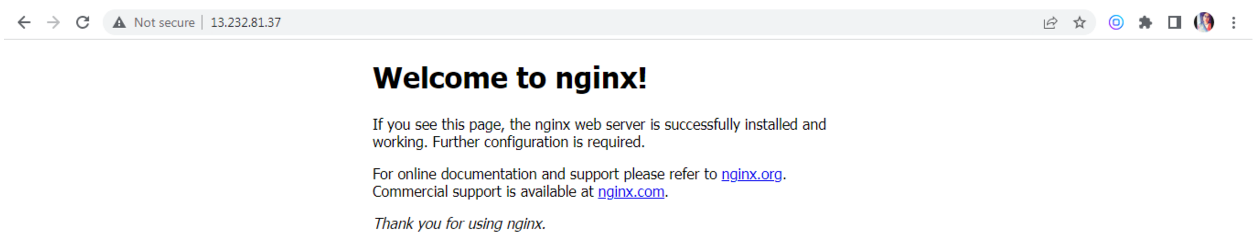


```
  System load:  0.0               Processes:             98
  Usage of /:   25.6% of 7.57GB   Users logged in:       0
  Memory usage: 24%               IPv4 address for eth0: 172.31.35.43
  Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

31 updates can be applied immediately.
17 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-172-31-35-43:~$
```

i-0dbebc151cdee5247 (myec2server)

PublicIPs: 13.232.81.37   PrivateIPs: 172.31.35.43

Step 14: Now open the EC2 instance. Copy the public IPv4 address and open it in a new tab. We can see that nginx server is working.



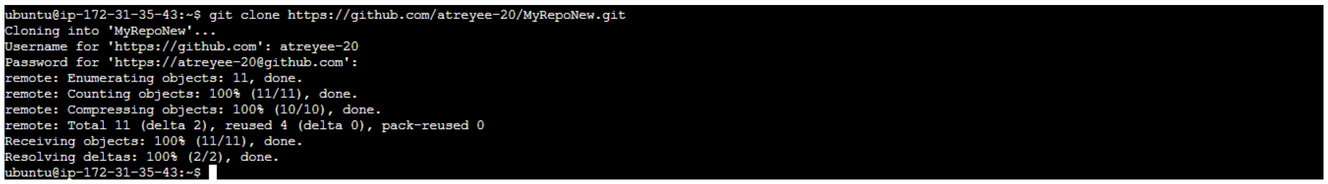
Welcome to nginx!

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

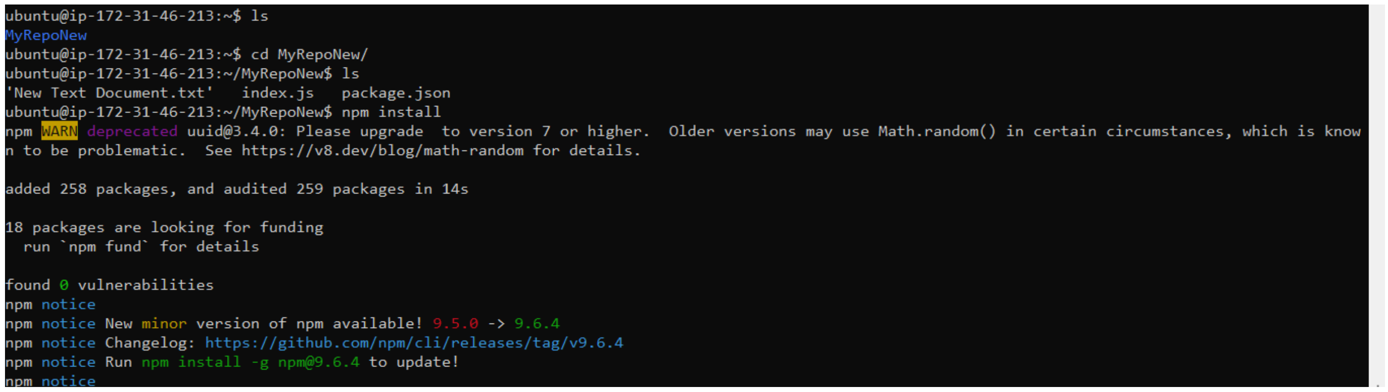
For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

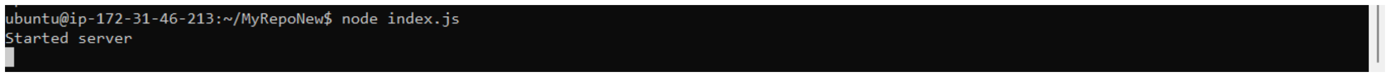
Step 15: Now go to the Connect terminal, clone your repository. Then open your repository, and install npm. Then start the server. (using the command as shown in Assignment no. 9).



```
ubuntu@ip-172-31-35-43:~$ git clone https://github.com/atreyee-20/MyRepoNew.git
Cloning into 'MyRepoNew'...
Username for 'https://github.com': atreyee-20
Password for 'https://atreyee-20@github.com':
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 11 (delta 2), reused 4 (delta 0), pack-reused 0
Receiving objects: 100% (11/11), done.
Resolving deltas: 100% (2/2), done.
ubuntu@ip-172-31-35-43:~$
```

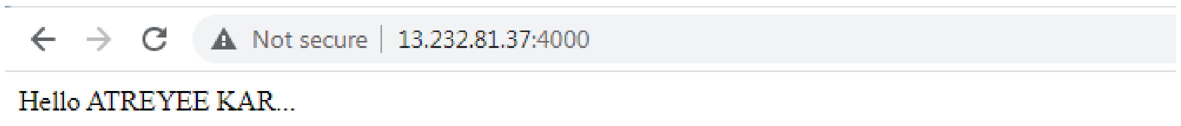


```
ubuntu@ip-172-31-46-213:~$ ls
MyRepoNew
ubuntu@ip-172-31-46-213:~$ cd MyRepoNew/
ubuntu@ip-172-31-46-213:~/MyRepoNew$ ls
'New Text Document.txt'   index.js   package.json
ubuntu@ip-172-31-46-213:~/MyRepoNew$ npm install
npm WARN deprecated uuid@3.4.0: Please upgrade  to version 7 or higher.  Older versions may use Math.random() in certain circumstances, which is know
n to be problematic.  See https://v8.dev/blog/math-random for details.

added 258 packages, and audited 259 packages in 14s

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

found 0 vulnerabilities
npm notice
npm notice New minor version of npm available! 9.5.0 -> 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-172-31-46-213:~/MyRepoNew$ node index.js
Started server
```

Step 16: Open the nginx page. Go to the IP address, put a colon and give the port number(In our case, it is 4000). We can see the content.



Hello ATREYEE KAR...

Thus, we have successfully deployed a project from GitHub to EC2 by creating a new Security group and user data.