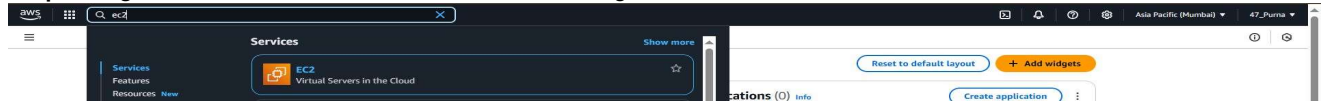
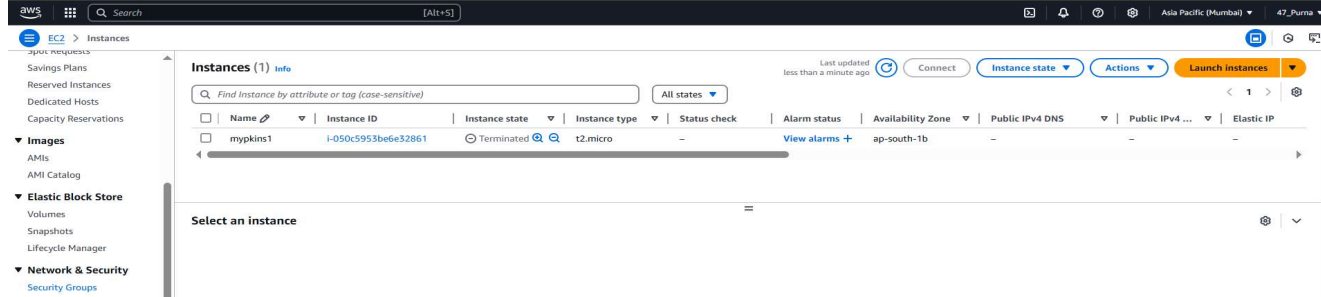


Assignment 10: Deploy a project from github to ec2 by creating new security group and user data.

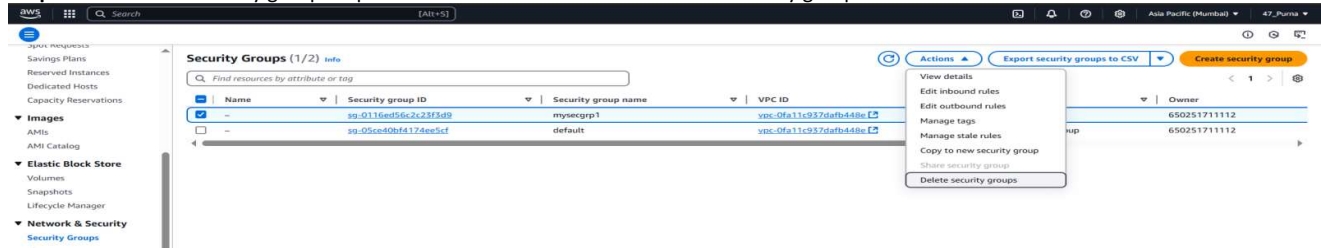
Step 1 : Log in to AWS account .To Launch an EC2 Instance Navigate to EC2 Dashboard.



Step 2: Go to security groups on EC2



Step3: Delete other security groups except the default one and click on create security group.



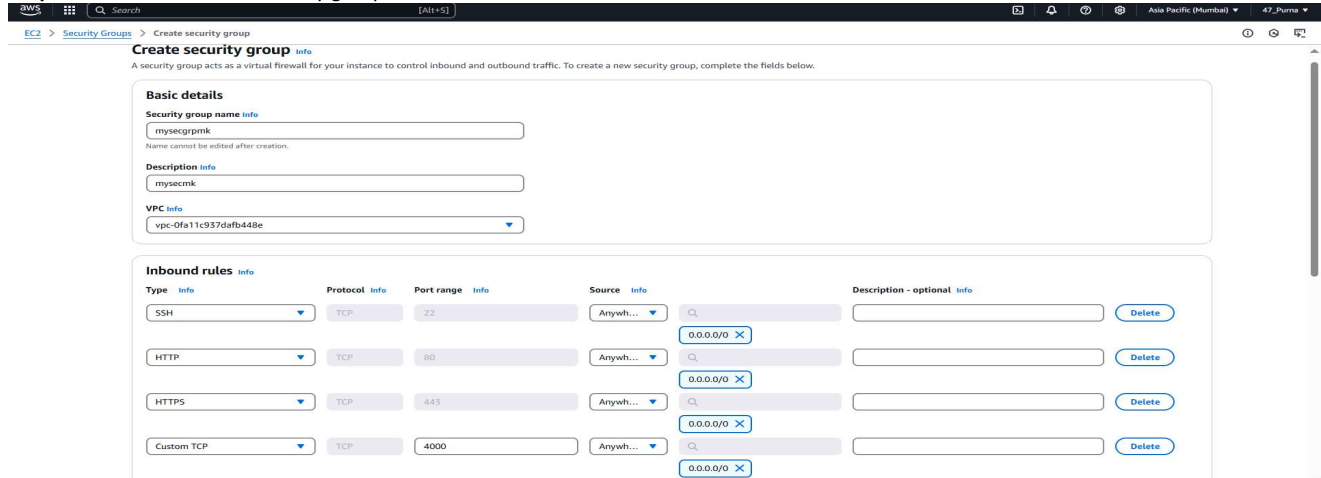
Step 4: click on Delete.



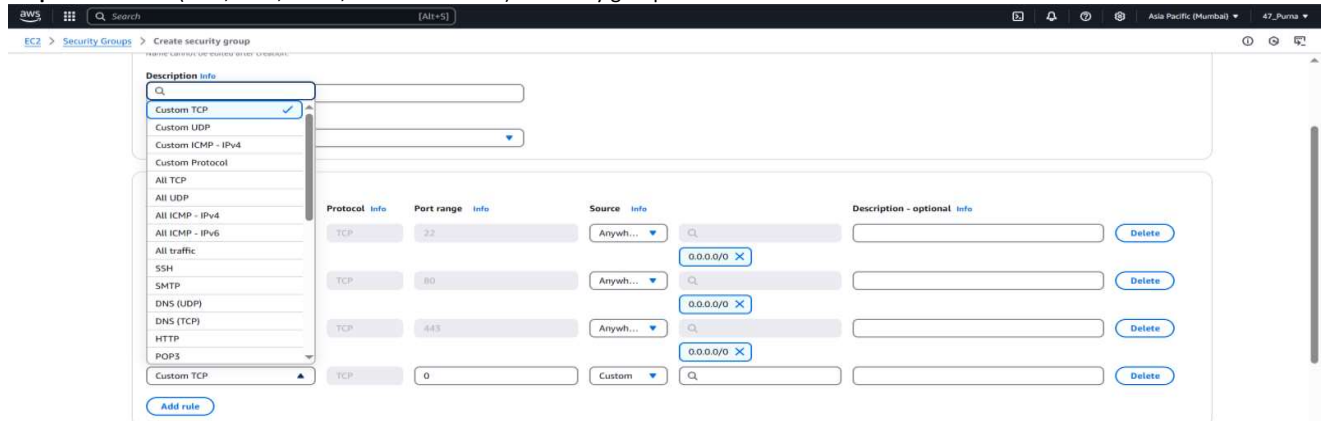
Step 5: Security group is successfully deleted. Now click on Create security group



Step6: Give a name to the security group and click on add inbound rule.



Step7: Add 4 rules(SSH,HTTP,HTTPS, and custom TCP)in security group.



Step8: Add port range as 4000 and source 0.0.0.0/0 .

Step9: click on Create security groups.

EC2 > Security Groups > Create security group

Description info

mysecgrp

VPC info

vpc-0fa11c937dafb448e

Inbound rules info

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

Outbound rules info

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

Tags - optional

No tags associated with the resource.

Create security group

Step10: Here we can see that security group is created .Now click on launch instances.

EC2 > Security Groups > sg-089df26c5aa1a7901 - mysecgrp

Details

Security group name: mysecgrp

Security group ID: sg-089df26c5aa1a7901

Description: mysecgrp

VPC ID: vpc-0fa11c937dafb448e

Inbound rules (4)

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source	Description
-	sg-0f07d6d47f0b544	IPv4	Custom TCP	TCP	4000	0.0.0.0/0	-
-	sg-0e21156a78f0557eb	IPv4	HTTP	TCP	80	0.0.0.0/0	-
-	sg-0eb6209b81a12c8b4	IPv4	HTTPS	TCP	443	0.0.0.0/0	-
-	sg-0c95d369c6e510a4c	IPv4	SSH	TCP	22	0.0.0.0/0	-

Step11: Give a name to the instance and select ubuntu.

EC2 > Instances > Launch an instance

Launch an instance

Name and tags

myminis

Application and OS Images (Amazon Machine Image)

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

Quick Start

Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE Linux, Debian

Summary

Number of instances: 1

Software Image (AMI): Canonical, Ubuntu, 24.04, amd64, read more

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 of opening an AWS account, you get 750 hours per month of t2.micro instance usage for 13.micro where t2.micro isn't available when used with free tier AMIs. 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GiB of snapshots, and 100 GiB of bandwidth to the internet.

Step 12: Create or select a key pair and existing security group . Click on security group which create earlier (here: mysecgrp).

EC2 > Instances > Launch an instance

Key pair name - required

mk_key

Network settings

Network

vpc-0fa11c937dafb448e

Subnet

No preference (Default subnet in any availability zone)

Auto-assign public IP

Enable

Firewall (security groups)

Create security group

Common security groups

sg-05ce40bf4174ae5cf

sg-089df26c5aa1a7901

Summary

Number of instances: 1

Software Image (AMI): Canonical, Ubuntu, 24.04, amd64, read more

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 of opening an AWS account, you get 750 hours per month of t2.micro instance usage for 13.micro where t2.micro isn't available when used with free tier AMIs. 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GiB of snapshots, and 100 GiB of bandwidth to the internet.

Launch instance

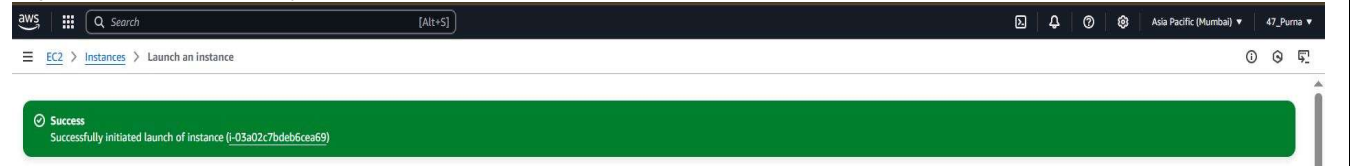
Step13: Set the Network settings part like this and go to Advanced details.

The screenshot shows the 'Launch Instance' page in the AWS Management Console, specifically the 'Network settings' section. The 'Key pair name' is set to 'mk_key'. Under 'Network', the 'vpc' is 'vpc-0fa11c937dafb448e' and the 'Subnet' is 'No preference (Default subnet in any availability zone)'. The 'Auto-assign public IP' is set to 'Enable'. Under 'Firewall (security groups)', the 'Create security group' option is selected. The 'Common security groups' section shows 'mysecgrp' and 'sg-089df26c5aa1a7901' selected. The 'Summary' panel on the right shows 'Number of instances' as 1, 'Software Image (AMI)' as 'Canonical, Ubuntu, 24.04, amd64...read more', 'Virtual server type (instance type)' as 't2.micro', 'Firewall (security group)' as 'mysecgrp', and 'Storage (volumes)' as '1 volume(s) - 8 GiB'. A 'Free tier' notification is visible, stating that in the first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet. The 'Launch instance' button is highlighted.

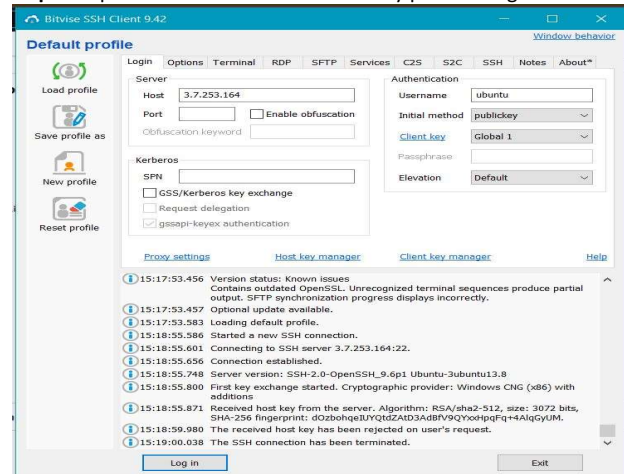
Step14: Write the code on user data portion and click on launch instance.

The screenshot shows the 'Launch Instance' page in the AWS Management Console, specifically the 'User data' section. The 'User data' field is populated with a script that sets up a V2 token, installs nginx, and sets up a Node.js environment. The 'Summary' panel on the right shows the same configuration as in Step 13. The 'Launch instance' button is highlighted.

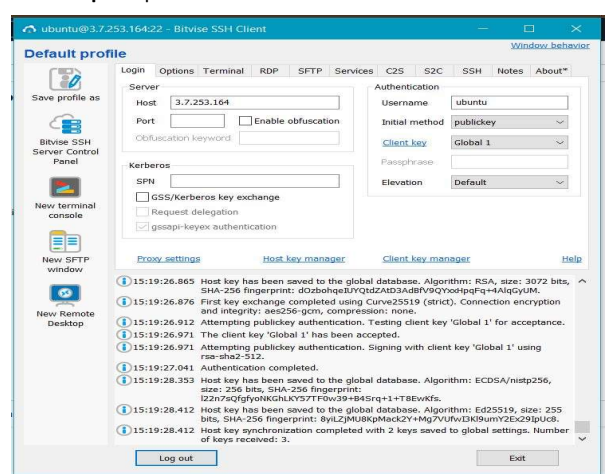
Step 14: Instance is successfully launched.



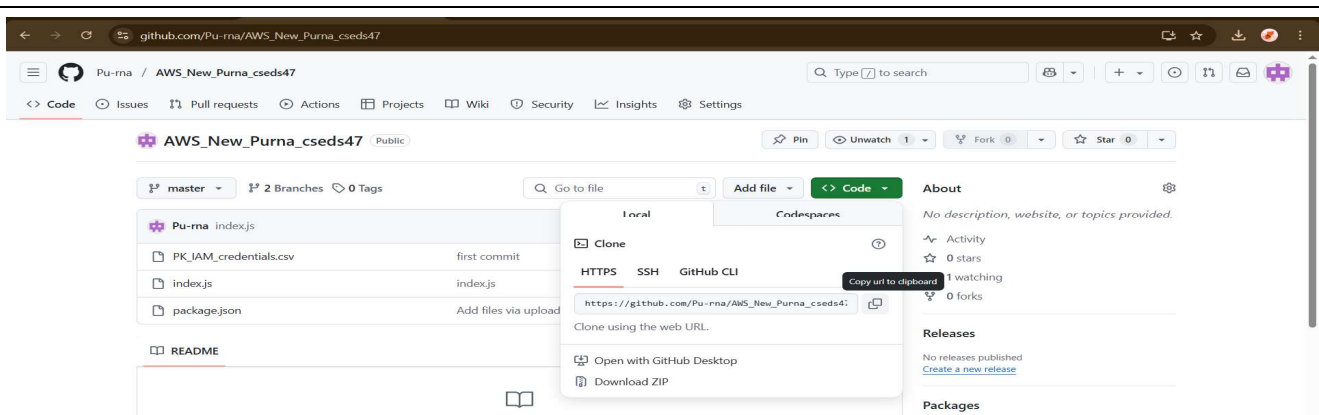
Step15: Open bitwise SSH client and set key pair and login.



Step16: open new terminal console.



Step 17: Open github repository and copy the link of the code.



Step18: Perform the following commands in the terminal.

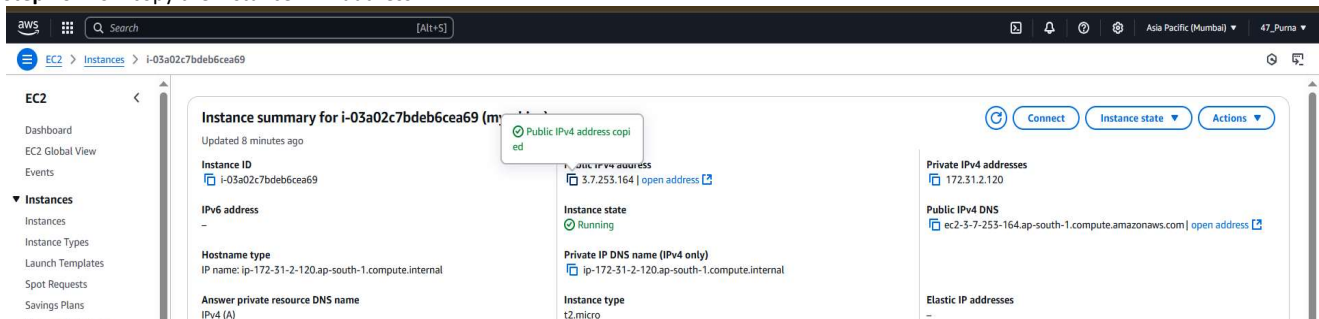
```
ubuntu@ip-172-31-2-120:~$ node -v
v18.20.8
ubuntu@ip-172-31-2-120:~$ git clone https://github.com/Pu-rna/AWS_New_Purna_cseds47.git
Cloning into 'AWS_New_Purna_cseds47'...
remote: Enumerating objects: 21, done.
remote: Counting objects: 100% (21/21), done.
remote: Compressing objects: 100% (21/21), done.
remote: Total 21 (delta 5), reused 2 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (21/21), 6.91 KiB | 1.73 MiB/s, done.
Resolving deltas: 100% (5/5), done.
ubuntu@ip-172-31-2-120:~$ ls
AWS_New_Purna_cseds47
```

```
ubuntu@ip-172-31-2-120:~$ cd AWS_New_Purna_cseds47
ubuntu@ip-172-31-2-120:~/AWS_New_Purna_cseds47$ ls
PK_IAM_credentials.csv  index.js  package.json
```

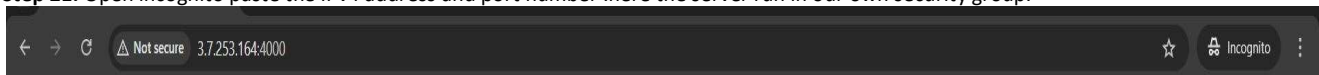
Step 19: To connect repository file of Github with the Ec2 pass this command and server is started.

```
ubuntu@ip-172-31-2-120:~/AWS_New_Purna_cseds47$ npm install
added 227 packages, and audited 228 packages in 15s
25 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
npm notice
npm notice New major version of npm available! 10.8.2 -> 11.2.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v11.2.0
npm notice To update run: npm install -g npm@11.2.0
npm notice
ubuntu@ip-172-31-2-120:~/AWS_New_Purna_cseds47$ node index.js
Started server
```

Step 20: Now copy the instance IPv4 address.



Step 21: Open incognito paste the IPv4 address and port number .here the server run in our own security group.



Hi Engineers