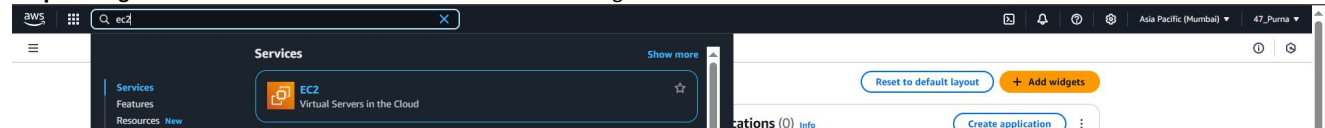
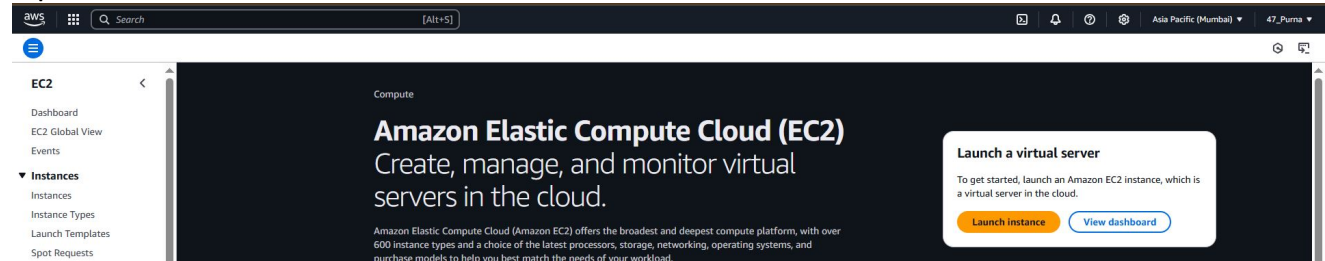


## Assignment 9: Deploy a project from Github to EC2.

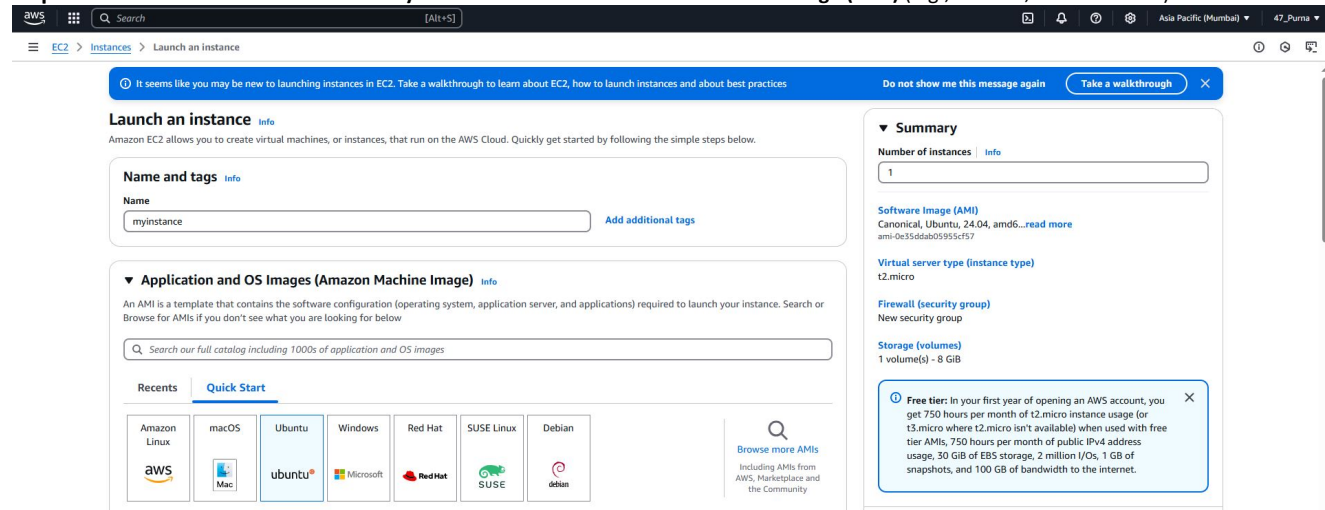
**Step 1 :** Log in to AWS account .To Launch an EC2 Instance Navigate to EC2 Dashboard.



**Step 2:** Click Launch Instance.



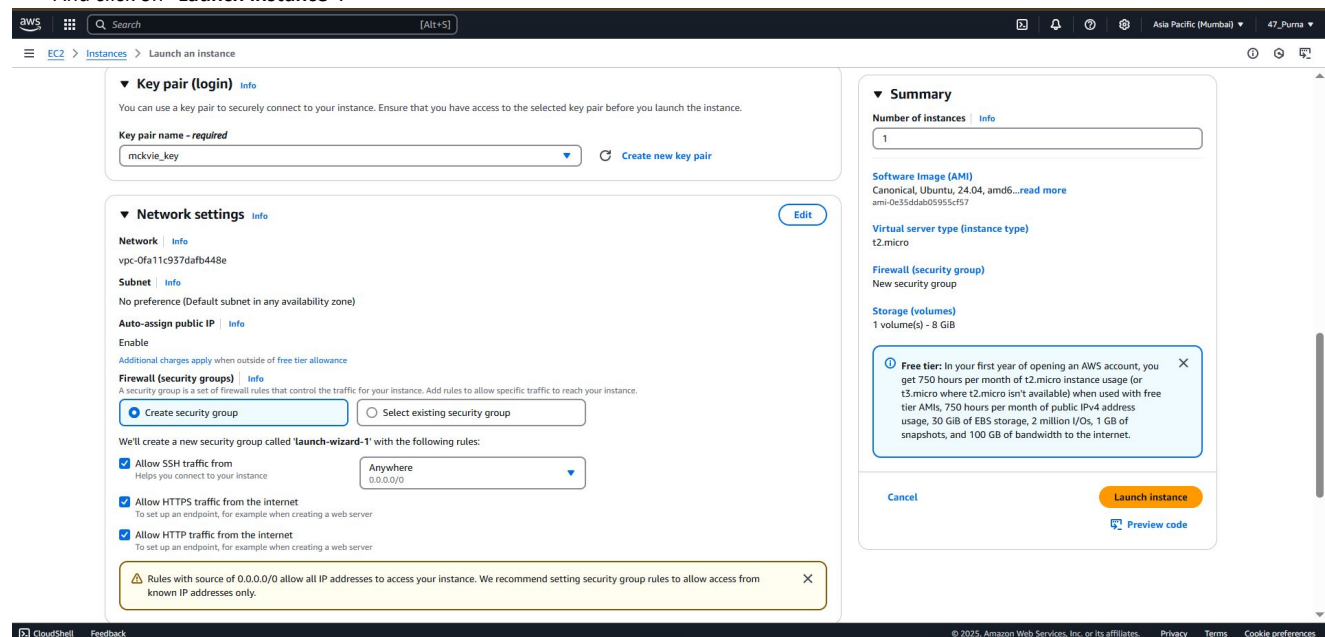
**Step 3:** Give a name of the instance as “myinstance”. Choose an Amazon Machine Image (AMI) (e.g., Ubuntu, Amazon Linux).



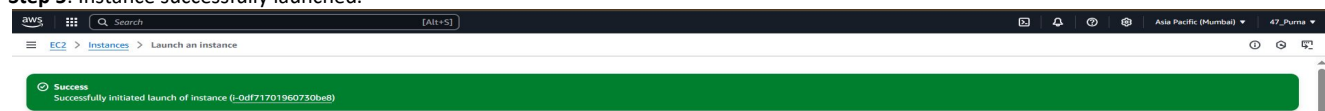
**Step 4:** Choose a key pair or create a key pair. To Configure Security Group:

- Allow SSH (Port 22) from IP.
- To hosting a web app, allow HTTP (Port 80) or HTTPS (Port 443).

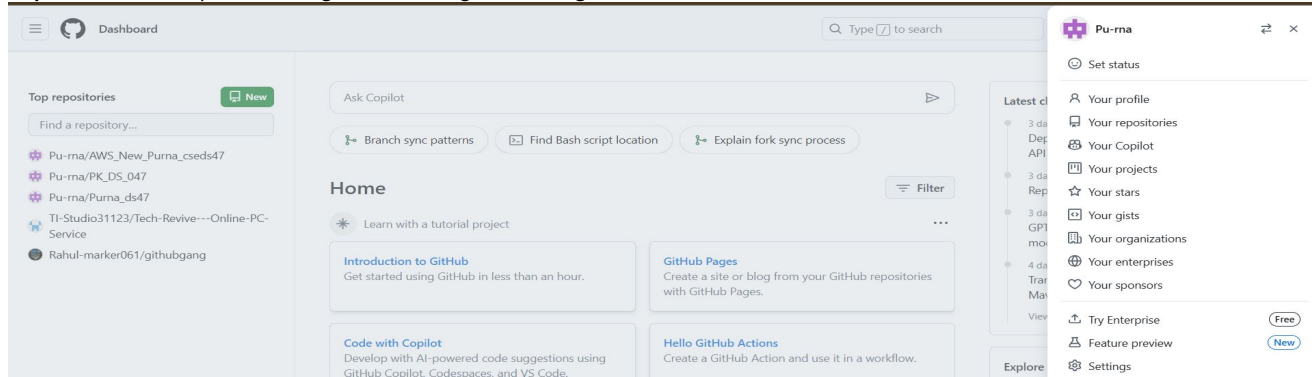
And click on “Launch instance”.



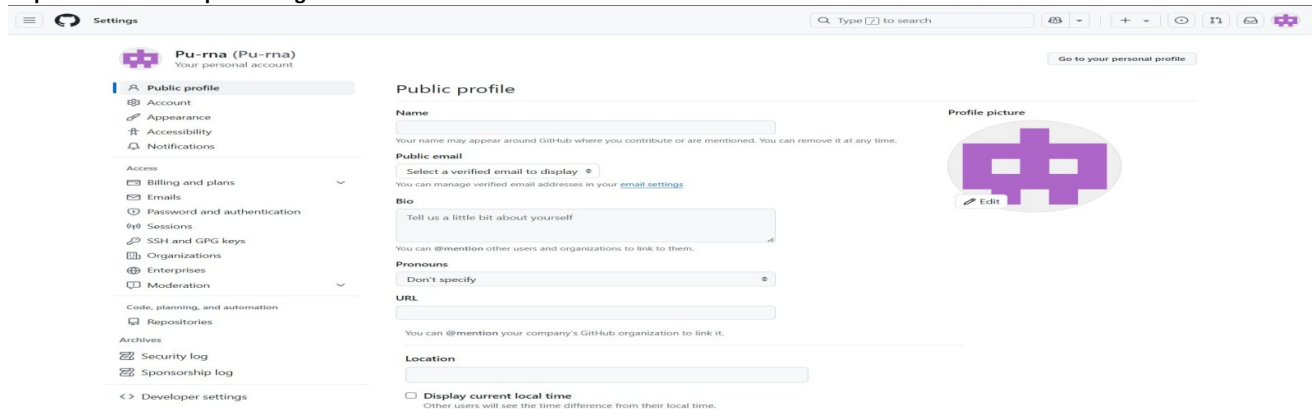
**Step 5:** Instance successfully launched.



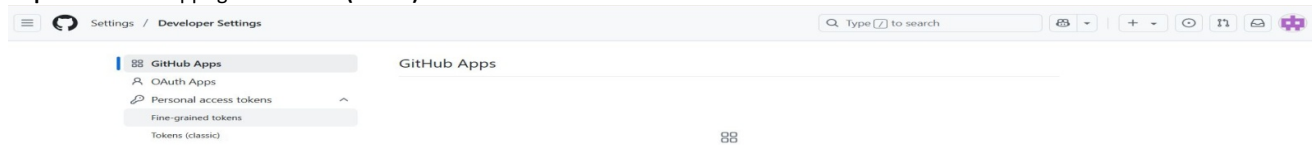
**Step 6: Go to GitHub profile and log in. After that go to “settings”.**



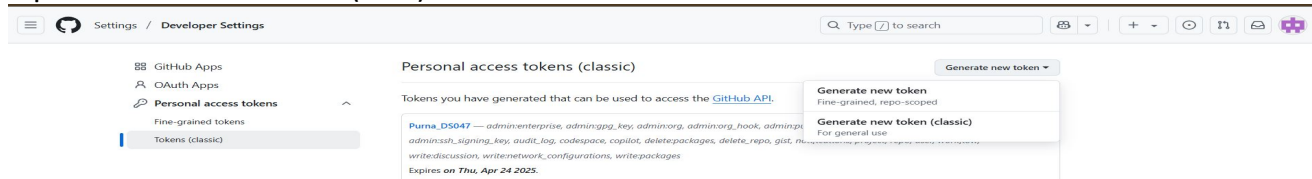
**Step 7: Go to “Developer settings”.**



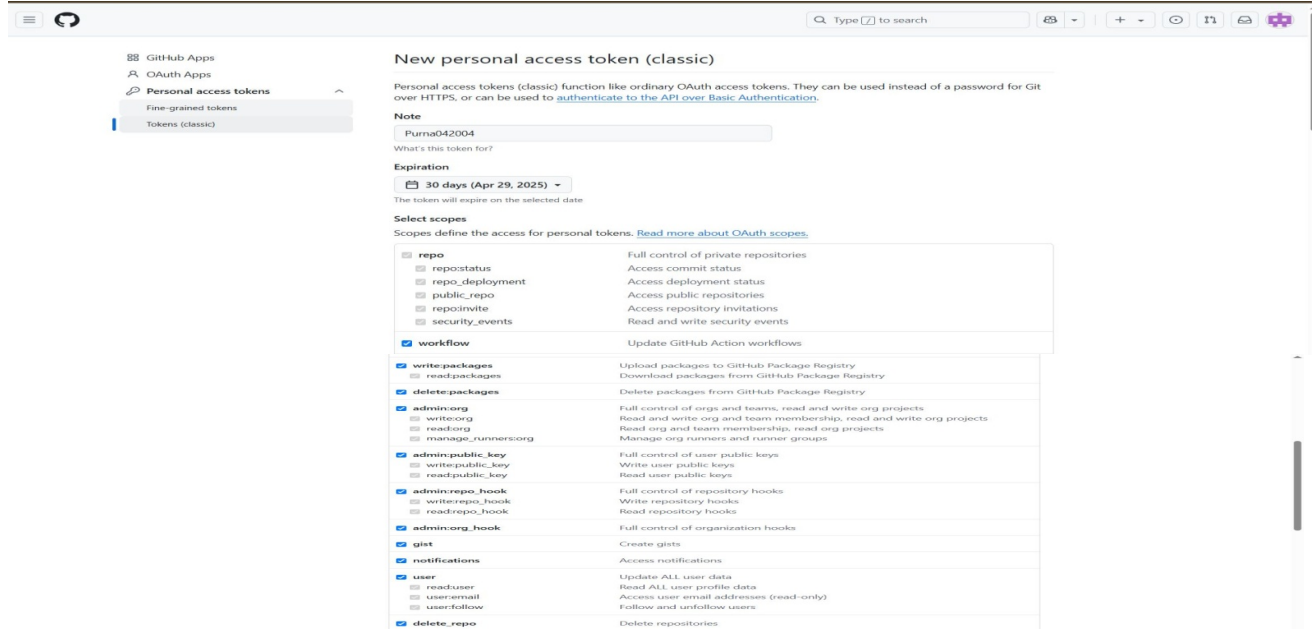
**Step 8: In GitHub Apps go to “Tokens(classic)” under Personal access tokens.**



**Step 9: Click on “Generate new token(classic)” under Generate new token.**



**Step 10: Give a Note as GitHub password which is accessible for 30 days. And check all the Select scopes. And click on “Generate token”.**



☒ admin:ssh\_signing\_key Full control of public user SSH signing keys
 ☐ writessh\_signing\_key Write public user SSH signing keys
 ☐ readssh\_signing\_key Read public user SSH signing keys

Generate token Cancel

Step 11: Token is generated . Now copy the URL.

Settings / Developer Settings
 

Type [Z] to search

Some of the scopes you've selected are included in other scopes. Only the minimum set of necessary scopes has been saved.

GitHub Apps
 OAuth Apps
 Personal access tokens
 Fine-grained tokens
 Tokens (classic)

Personal access tokens (classic)
 Generate new token
 Tokens you have generated that can be used to access the [GitHub API](#).
 Make sure to copy your personal access token now. You won't be able to see it again!
 

ghp\_8WZBRUjv1PYAm0hyu0LkYbfTUr17I25H0dpt
 Delete

Purna\_DS047 — admin:enterprise, admin:pgp\_key, admin:org, admin:org\_hook, admin:public\_key, admin:repo\_hook, admin:ssh\_signing\_key, audit\_log, codespace, copilot, delete:packages, delete:repo, gist, notifications, project, repo, user, workflow, write:discussion, write:network\_configurations, write:packages
 Never used
 Delete

Expires on Thu, Apr 24 2025.

Step 12: Now go to AWS site and click into instance ID.

AWS
 Search
 [Alt+S]

EC2 > Instances

Dashboard
 EC2 Global View
 Events

Instances
 Instances

Instances (1) Info
 Find Instance by attribute or tag (case-sensitive)
 All states

Last updated less than a minute ago
 Connect
 Instance state
 Actions
 Launch instances

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
<input type="checkbox"/>	myinstance	i-0df71701960730be8	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1b	ec2-13-232-139-164.ap...	13.232.139.164	-

Step 13: From instance details copy the Public IPv4 address.

AWS
 Search
 [Alt+S]

EC2 > Instances > i-0df71701960730be8

Dashboard
 EC2 Global View
 Events

Instances
 Instance Types
 Launch Templates
 Spot Requests
 Savings Plans
 Reserved Instances
 Dedicated Hosts
 Capacity Reservations

Images
 AMIs
 AMI Catalog

Elastic Block Store
 Volumes
 Snapshots
 Lifecycle Manager

Instance summary for i-0df71701960730be8 (m)
 Updated less than a minute ago
 Public IPv4 address copied

Instance ID
 i-0df71701960730be8
 IP v6 address
 -
 Hostname type
 IP name: ip-172-31-12-123.ap-south-1.compute.internal
 Answer private resource DNS name
 IPv4 (a)
 Auto-assigned IP address
 ec2-13-232-139-164 [Public IP]
 IAM Role
 -
 IMDSv2
 Required
 Operator
 -

Private IP DNS name (IPv4 only)
 ip-172-31-12-123.ap-south-1.compute.internal
 Instance type
 t2.micro
 VPC ID
 vpc-0fa11c937dafb448e
 Subnet ID
 subnet-05d3a96b79f214073
 Instance ARN
 arn:aws:ec2:ap-south-1:65025171112:instance/i-0df71701960730be8

Private IPv4 addresses
 172.31.12.123
 Public IPv4 DNS
 ec2-13-232-139-164.ap-south-1.compute.amazonaws.com | open address
 Elastic IP addresses
 -
 AWS Compute Optimizer finding
 It is taking a bit longer than usual to fetch your data
 Auto Scaling Group name
 -
 Managed
 false

Step 14: Now open Bitvise SSH and paste the ID in Server:Host.

Bitvise SSH Client 9.42

Default profile
 Login Options Terminal RDP SFTP Services C2S S2C SSH Notes About\*

Load profile
 Save profile as
 New profile
 Reset profile

Server
 Host
 13.232.139.164
 Port
 22
 Enable obfuscation
 Obfuscation keyword
 Authentication
 Username
 ubuntu
 Initial method
 publickey
 Client key
 Global 1
 Passphrase
 -
 Elevation
 Default
 Kerberos
 SPN
 -
 GSS/Kerberos key exchange
 Request delegation
 gssapi-keyex authentication
 Proxy settings
 Host key manager
 Client key manager
 Help

Log in
 Exit

Step 15: Now click on Client key manager and import key pair which is use to launched instance. After tha close the tab.

Bitvise Client Key Management | Cryptographic provider: Windows CNG (x86) with additions

Client Key Manager
 You have the following SSH user authentication keys:
 Location
 Algorithm
 Size
 SHA-256 Fingerprint
 MD5 Fingerprint
 Bubble Babble
 Comment
 Client keys supported by the current crypto provider (1):
 Global 1
 RSA
 2048
 no
 YPkeH0QdWgrXVdW11e4...
 d738-69:7c9:18:...
 xunw-fimod-velhyz-...

Generate New
 Modify
 Remove
 Import
 Export
 Change Passphrase
 More

12:19:43.398 Automatic check for updates completed successfully.

Step 16: Now click on log in. And open New terminal console.

ubuntu@13.232.139.164:22 - Bitvise SSH Client

Default profile
 Login Options Terminal RDP SFTP Services C2S S2C SSH Notes About\*

Save profile as
 Bitvise SSH Server Control Panel
 New terminal console
 New SFTP window
 New Remote Desktop

Server
 Host
 13.232.139.164
 Port
 22
 Enable obfuscation
 Obfuscation keyword
 Authentication
 Username
 ubuntu
 Initial method
 publickey
 Client key
 Global 1
 Passphrase
 -
 Elevation
 Default
 Kerberos
 SPN
 -
 GSS/Kerberos key exchange
 Request delegation
 gssapi-keyex authentication
 Proxy settings
 Host key manager
 Client key manager
 Help

Log out
 Exit

Purna Kundu/DS/22/047



```
ubuntu@ip-172-31-12-123:~$ sudo apt-get install update
```

```
ubuntu@ip-172-31-12-123:~$ sudo apt-get install upgrade
```

```
ubuntu@ip-172-31-12-123:~$ sudo apt-get install nginx
```

```
ubuntu@ip-172-31-12-123:~$ nginx -v
nginx version: nginx/1.24.0 (Ubuntu)
ubuntu@ip-172-31-12-123:~$
```

```
ubuntu@ip-172-31-12-123:~$ curl -sL https://deb.nodesource.com/setup_18.x | sudo -E bash -
2025-03-30 07:06:18 - Installing pre-requisites
Reading package lists... Done
2025-03-30 07:06:18 - Repository configured successfully.
2025-03-30 07:06:54 - To install Node.js, run: apt-get install nodejs -y
2025-03-30 07:06:54 - You can use N|solid Runtime as a node.js alternative
2025-03-30 07:06:54 - To install N|solid Runtime, run: apt-get install nsolid -y
```

```
ubuntu@ip-172-31-12-123:~$ sudo apt install nodejs
```

```
ubuntu@ip-172-31-12-123:~$ node -v
v18.20.8
```

The screenshot shows the GitHub repository page for 'AWS\_New\_Purna\_cseds47'. At the top, there's a navigation bar with links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below this, the repository name 'AWS\_New\_Purna\_cseds47' is displayed with a 'Public' badge. To the right of the repository name are buttons for Pin, Unwatch (1), Fork (0), Star (0), and a dropdown menu. Below the repository name, there's a section for the 'master' branch, showing '1 Branch' and '0 Tags'. A search bar is present with the text 'Go to file'. To the right of the search bar are buttons for 'Add file' and '<> Code'. Below this, a table lists the files in the repository: 'PK\_IAM\_credentials.csv' (last commit, last week), 'index.js' (index.js, 5 days ago), and 'package.json' (Add files via upload, 5 days ago). To the right of the file list, there's a section for 'About' with the text 'No description, website, or topics provided.' and a list of activity: 'Activity', '0 stars', '1 watching', and '0 forks'.

```
ubuntu@ip-172-31-12-123:~$ git clone https://github.com/Pu-rna/AWS_New_Purna_cseds47.git
Cloning into 'AWS_New_Purna_cseds47'..
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 10 (delta 2), reused 2 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (10/10), done.
Resolving deltas: 100% (2/2), done.
```

```
ubuntu@ip-172-31-12-123:~$ ls
AWS_New_Purna_cseds47
ubuntu@ip-172-31-12-123:~$ cd AWS_New_Purna_cseds47
ubuntu@ip-172-31-12-123:~/AWS_New_Purna_cseds47$ ls
PK_IAM_credentials.csv  index.js  package.json
ubuntu@ip-172-31-12-123:~/AWS_New_Purna_cseds47$ npm install

added 227 packages, and audited 228 packages in 11s

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

found 0 vulnerabilities
ubuntu@ip-172-31-12-123:~/AWS_New_Purna_cseds47$ node index.js
Started server
```

**EC2** > Instances

Dashboard  
EC2 Global View

### Instances

Find instance by attribute or tag (case-sensitive) [All states]

Name	Instance ID	Instance state	Instance type	Status checks	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
myinstance	i-0df71701960730be8	Running	t2.micro	2/2 checks pass	View alarms +	ap-south-1b	ec2-13.232.139-164.ap-south-1.compute.amazonaws.com	13.232.139.164	-

Last updated less than a minute ago

[Connect] [Instance state] [Actions] [Launch instances]

---

#### i-0df71701960730be8 (myinstance)

[Details] [Status and alarms] [Monitoring] [Security] [Networking] [Storage] [Tags]

**Instance summary**

**Instance ID**  
i-0df71701960730be8

**IPv4 address**  
-

**Hostname type**  
IP name: ip-172-31-12-123.ap-south-1.compute.internal

**Answer private resource DNS name**  
IPv4 (A)  
-

**Auto-assigned IP address**  
13.232.139.164 [Public IP]

**IAM Role**  
-

**Public IPv4 address copy ed**

ec2-13.232.139-164.ap-south-1.compute.amazonaws.com [open address]

**Instance state**  
Running

**Private IP DNS name (IPv4 only)**  
ip-172-31-12-123.ap-south-1.compute.internal

**Instance type**  
t2.micro

**VPC ID**  
vpc-0fa11c937dafb448e

**Subnet ID**  
subnet-05d3a96b79f214073

**Private IPv4 addresses**  
172.31.12.123

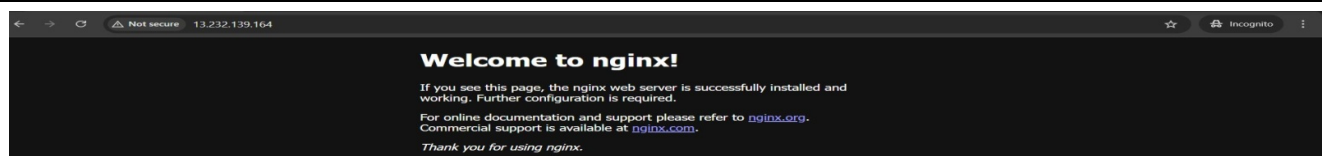
**Public IPv4 DNS**  
ec2-13.232.139-164.ap-south-1.compute.amazonaws.com [open address]

**Elastic IP addresses**  
-

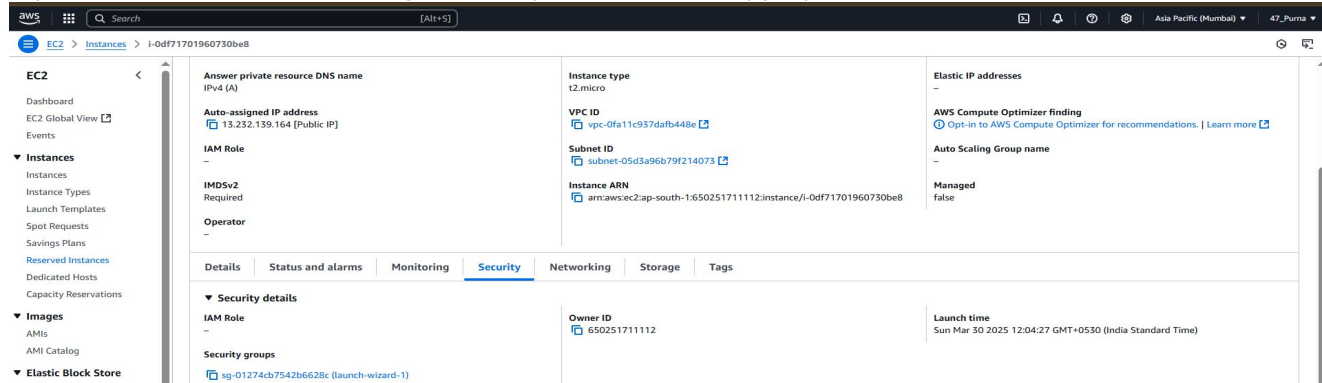
**AWS Compute Optimizer finding**  
Opt-in to AWS Compute Optimizer for recommendations. [Learn more]

**Auto Scaling Group name**  
-

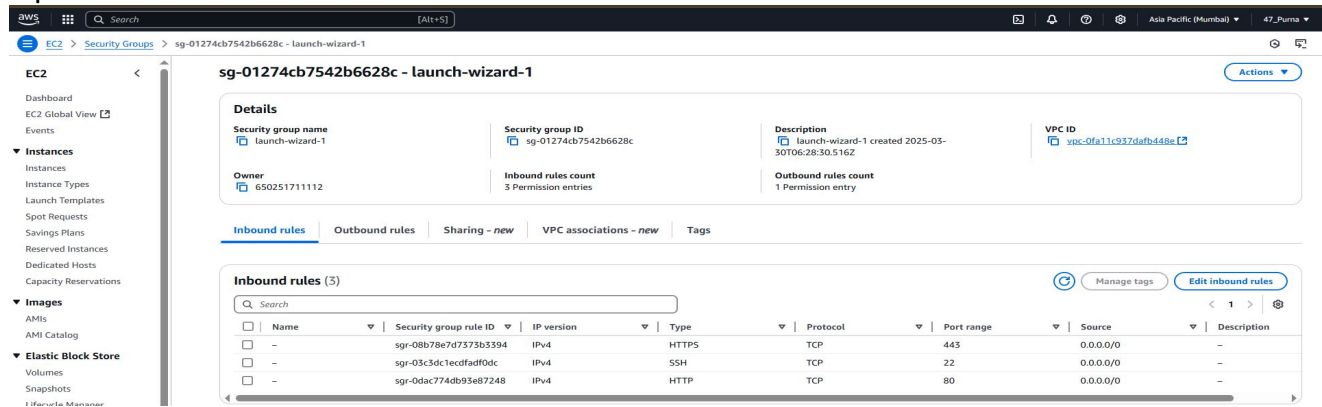
Purna Kundu/DS/22/047



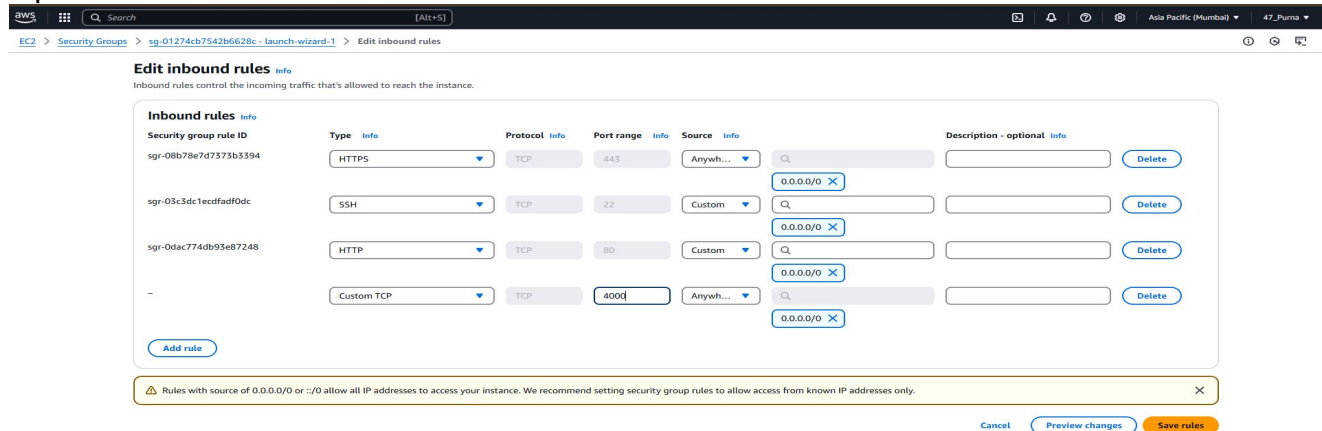
**Step 23:** Now click on instance id after that go to **“Security”** and click on **“Security groups”**.



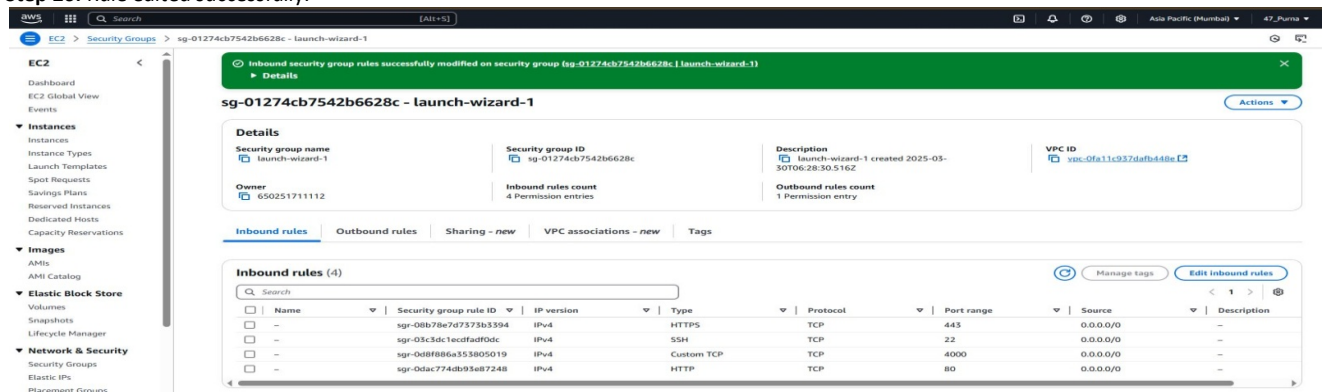
**Step 24:** Now click on **“Edit inbound rules”**.



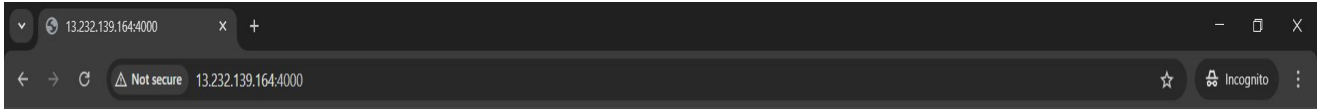
**Step 25:** Now edit inbound rules and Save rules.



**Step 26:** Rule edited successfully.

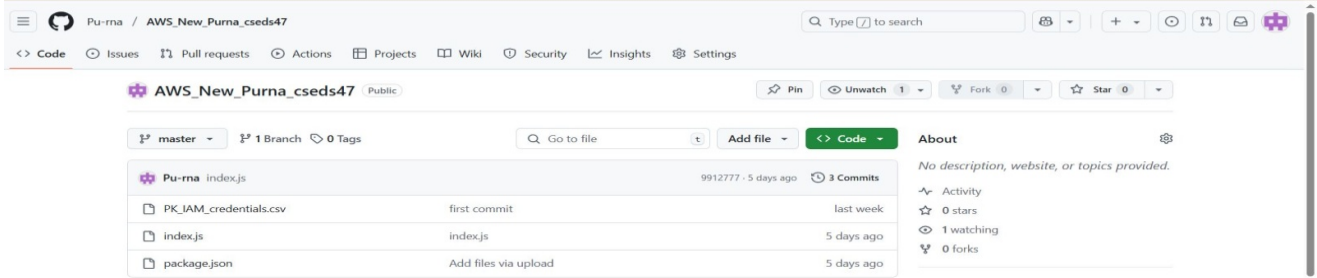


**Step 27:** Now again refresh the previous incognito tab and now the repository file is successfully accessible.

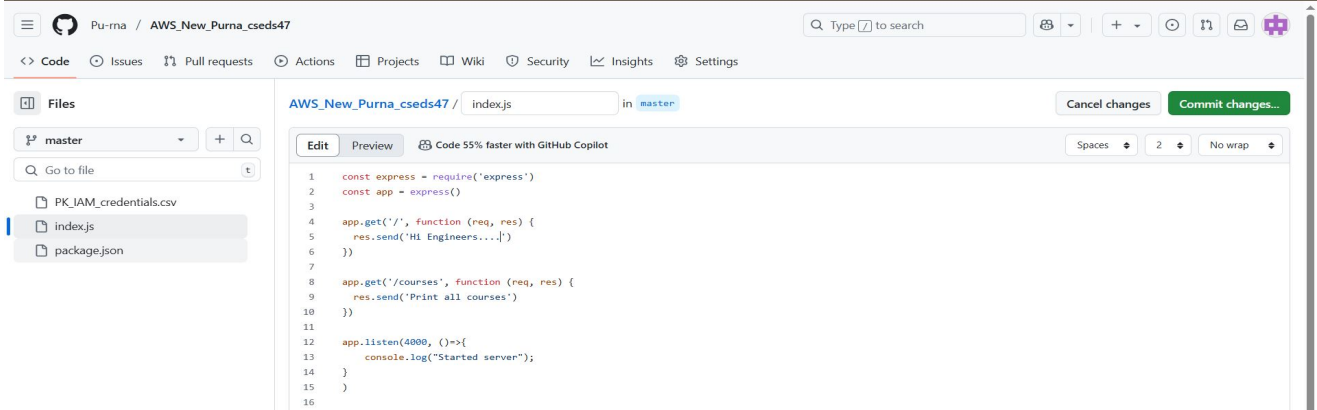


Hello MCKV Purna

**Step 28:** Now go to github repository and go to the file `index.js`.

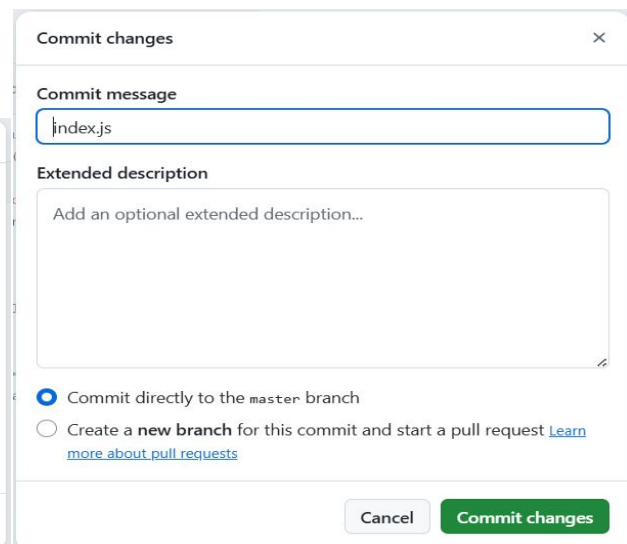
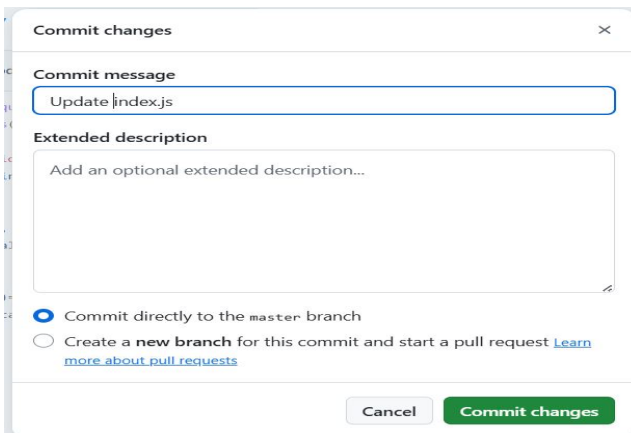


**Step 29:** Now edit the file and click on **commit changes..**



**Step 30:** Now this page is open delete **Update** from **commit message** .

**Step 31:** And click on **commit changes**.



**Step 32:** Now refresh the incognito mode and file is successfully edited.

