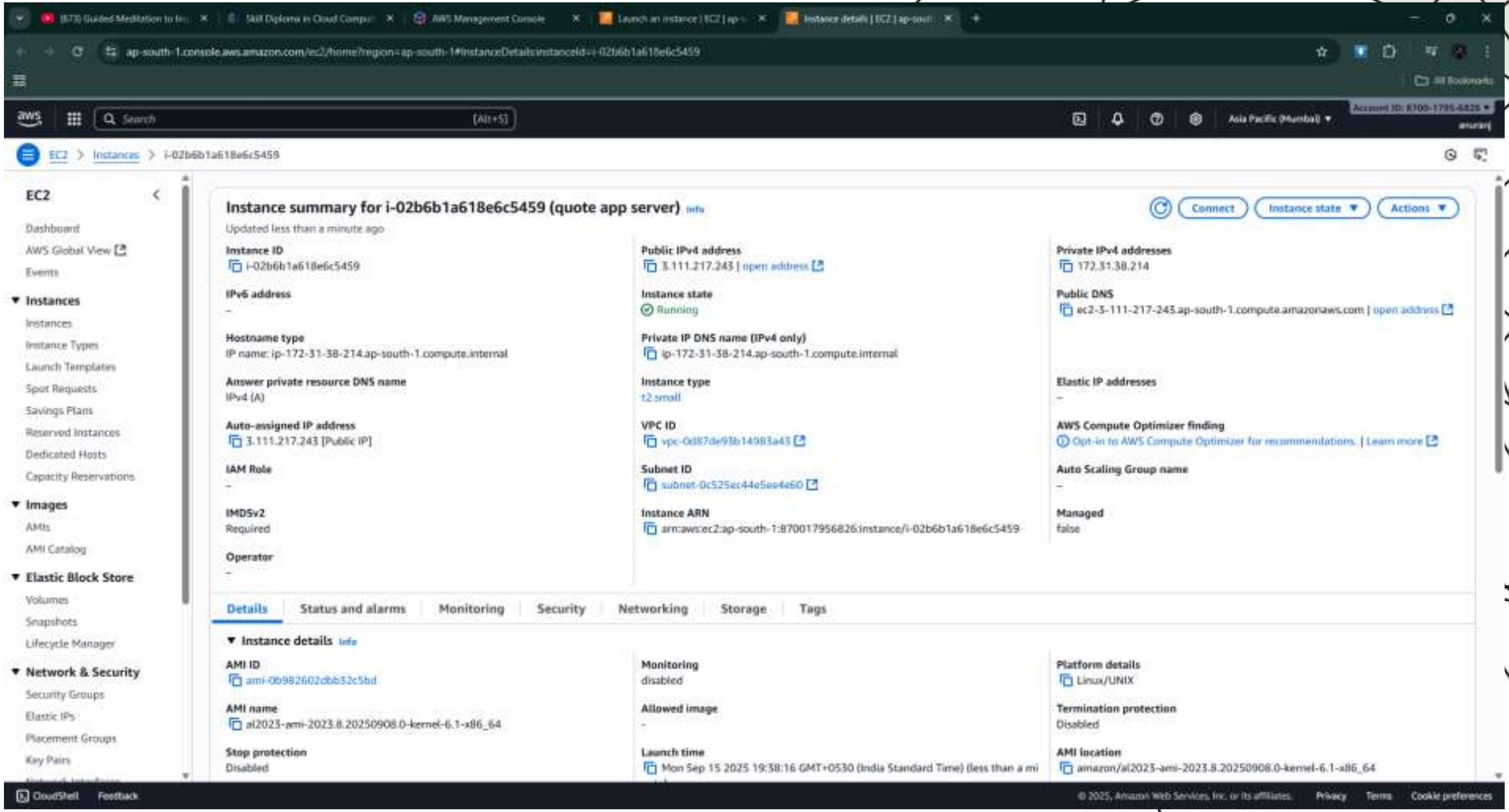
DEPLOY WEBSITE WITH DATABASE IN AWS

I started by creating an ec2 instance.i chose amazon linux for it



Instance summary for i-02b6b1a618e6c5459 (quote app server) Info

Updated less than a minute ago

Instance ID
i-02b6b1a618e6c5459

IPv6 address
–

Hostname type
IP name: ip-172-31-38-214.ap-south-1.compute.internal

Answer private resource DNS name
IPv4 (A)

Auto-assigned IP address
3.111.217.243 [Public IP]

IAM Role
–

IMDSv2
Required

Operator
–

Public IPv4 address
3.111.217.243 | open address

Instance state
Running

Private IP DNS name (IPv4 only)
ip-172-31-38-214.ap-south-1.compute.internal

Instance type
t2.small

VPC ID
vpc-0d87de93b14983a43

Subnet ID
subnet-0c525ec44e5ea4e60

Instance ARN
arn:aws:ec2:ap-south-1:870017956826:instance/i-02b6b1a618e6c5459

Private IPv4 addresses
172.31.38.214

Public DNS
ec2-3-111-217-243.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
–

Managed
false

▼ Instance details Info

AMI ID
ami-0b982602dbb32c5bd

AMI name
al2023-ami-2023.8.20250908.0-kernel-6.1-x86_64

Stop protection
Disabled

Monitoring
disabled

Allowed image
-

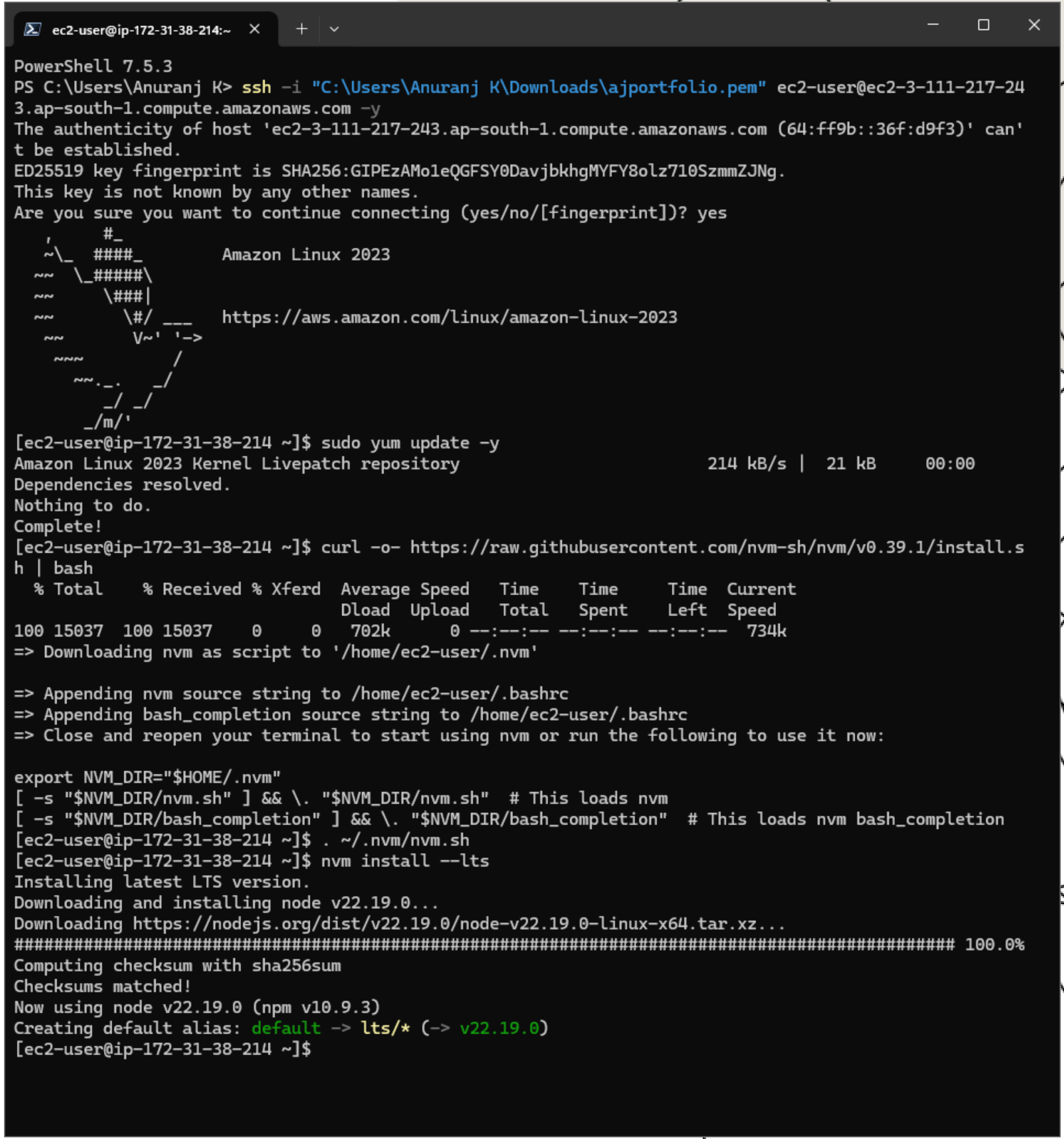
Launch time
Mon Sep 15 2025 19:38:16 GMT+0530 (India Standard Time) (less than a mi

Platform details
Linux/UNIX

Termination protection
Disabled

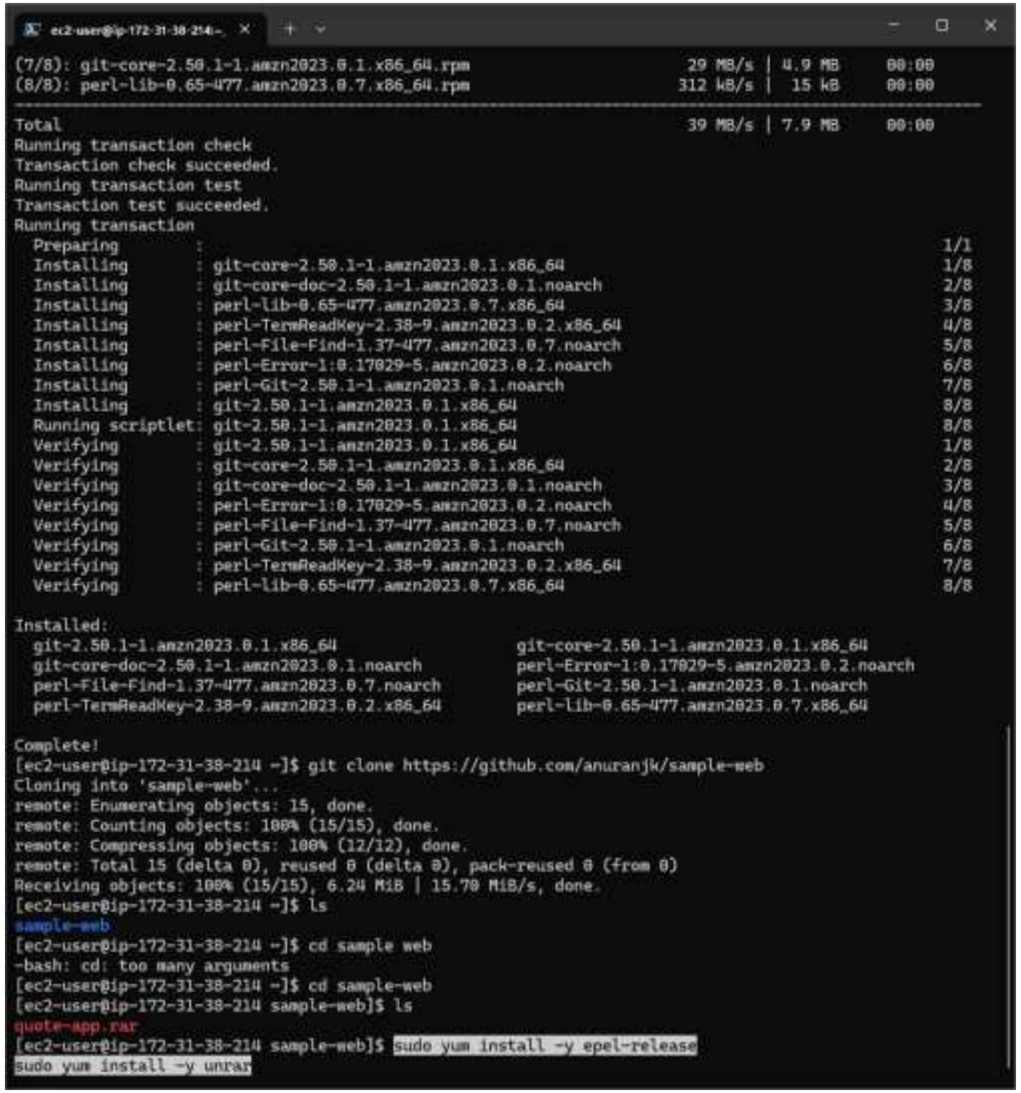
AMI location
amazon/al2023-ami-2023.8.20250908.0-kernel-6.1-x86_64

Then I logined into the server
Then I installed node.js and npm so inorder to install that properly I used nvm (node version manager. >sudo yum update -y
 > curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.1/install.sh | bash
Then I ran the script > . ~/.nvm/nvm.sh
Then I installed the most stable and latest version of node.js using this command > nvm install --lts

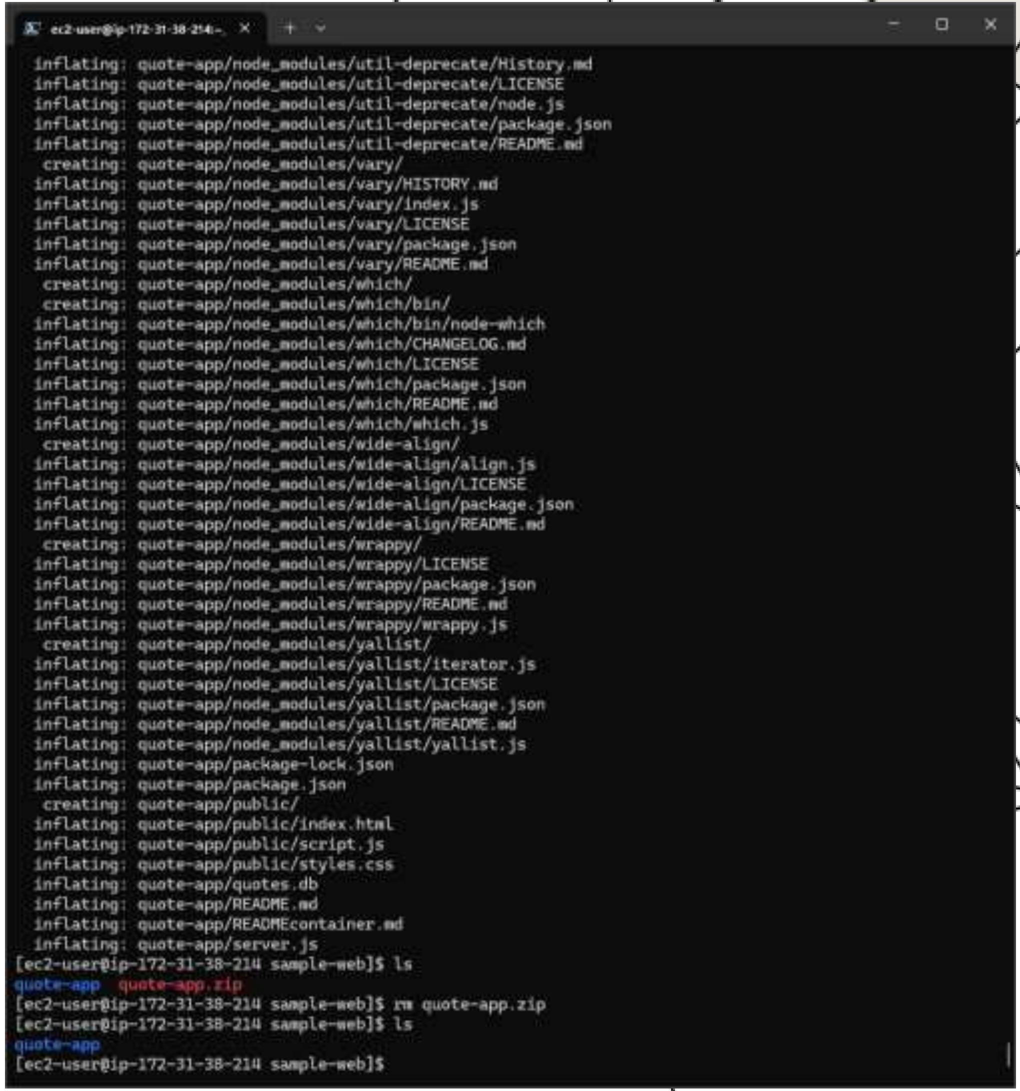


```
PowerShell 7.5.3
PS C:\Users\Anuranj K> ssh -i "C:\Users\Anuranj K\Downloads\ajportfolio.pem" ec2-user@ec2-3-111-217-24
3.ap-south-1.compute.amazonaws.com -y
The authenticity of host 'ec2-3-111-217-243.ap-south-1.compute.amazonaws.com (64:ff9b::36f:d9f3)' can'
t be established.
ED25519 key fingerprint is SHA256:GIPEzAMo1eQGFSY0DavjbkhgMYFY8olz710SzmmZJNg.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-38-214 ~]$ sudo yum update -y
Amazon Linux 2023 Kernel Livepatch repository                         214 kB/s |  21 kB     00:00
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-38-214 ~]$ curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.1/install.s
h | bash
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 15037  100 15037    0     0   702k      0 --:--:-- --:--:-- --:--:--  734k
=> Downloading nvm as script to '/home/ec2-user/.nvm'

=> Appending nvm source string to /home/ec2-user/.bashrc
=> Appending bash_completion source string to /home/ec2-user/.bashrc
=> Close and reopen your terminal to start using nvm or run the following to use it now:

export NVM_DIR="$HOME/.nvm"
[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh"  # This loads nvm
[ -s "$NVM_DIR/bash_completion" ] && \. "$NVM_DIR/bash_completion"  # This loads nvm bash_completion
[ec2-user@ip-172-31-38-214 ~]$ . ~/.nvm/nvm.sh
[ec2-user@ip-172-31-38-214 ~]$ nvm install --lts
Installing latest LTS version.
Downloading and installing node v22.19.0...
Downloading https://nodejs.org/dist/v22.19.0/node-v22.19.0-linux-x64.tar.xz...
################################################################################################ 100.0%
Computing checksum with sha256sum
Checksums matched!
Now using node v22.19.0 (npm v10.9.3)
Creating default alias: default -> lts/* (-> v22.19.0)
[ec2-user@ip-172-31-38-214 ~]$
```

Then I tried cloning from my git repo but I faced some problems because the file was in rar file which cannot be extracted in amazon linux without addition tools but I main tools was no longer available for amazon linux so I uploaded a zip file of the source code and proceeded with it. I also find some problems while deleting the previous directory so this is the command I used > rmdir –rf dir_name

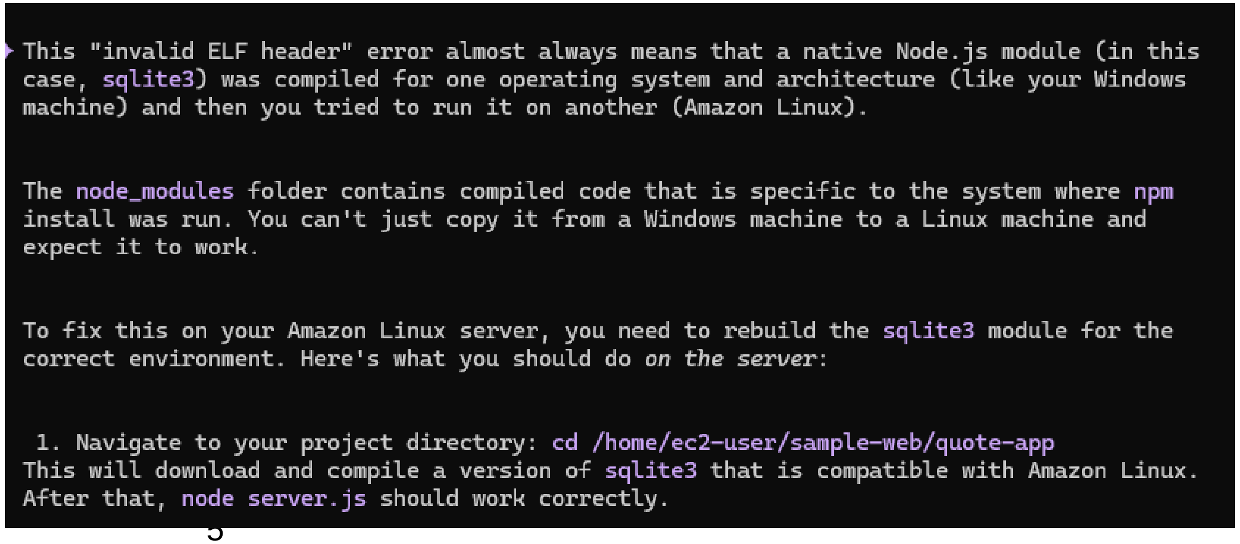


```
(7/8): git-core-2.50.1-1.amzn2023.0.1.x86_64.rpm                          29 MB/s | 4.9 MB     00:00
(8/8): perl-lib-0.65-477.amzn2023.0.7.x86_64.rpm                         312 kB/s |  15 kB     00:00
--------------------------------------------------------------------------------------------------------
Total                                                                      39 MB/s | 7.9 MB     00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                                                1/1
  Installing       : git-core-2.50.1-1.amzn2023.0.1.x86_64                                          1/8
  Installing       : git-core-doc-2.50.1-1.amzn2023.0.1.noarch                                      2/8
  Installing       : perl-lib-0.65-477.amzn2023.0.7.x86_64                                          3/8
  Installing       : perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64                                    4/8
  Installing       : perl-File-Find-1.37-477.amzn2023.0.7.noarch                                    5/8
  Installing       : perl-Error-1:0.17029-5.amzn2023.0.2.noarch                                     6/8
  Installing       : perl-Git-2.50.1-1.amzn2023.0.1.noarch                                          7/8
  Installing       : git-2.50.1-1.amzn2023.0.1.x86_64                                               8/8
  Running scriptlet: git-2.50.1-1.amzn2023.0.1.x86_64                                               8/8
  Verifying        : git-2.50.1-1.amzn2023.0.1.x86_64                                               1/8
  Verifying        : git-core-2.50.1-1.amzn2023.0.1.x86_64                                          2/8
  Verifying        : git-core-doc-2.50.1-1.amzn2023.0.1.noarch                                      3/8
  Verifying        : perl-Error-1:0.17029-5.amzn2023.0.2.noarch                                     4/8
  Verifying        : perl-File-Find-1.37-477.amzn2023.0.7.noarch                                    5/8
  Verifying        : perl-Git-2.50.1-1.amzn2023.0.1.noarch                                          6/8
  Verifying        : perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64                                    7/8
  Verifying        : perl-lib-0.65-477.amzn2023.0.7.x86_64                                          8/8

Installed:
  git-2.50.1-1.amzn2023.0.1.x86_64                     git-core-2.50.1-1.amzn2023.0.1.x86_64
  git-core-doc-2.50.1-1.amzn2023.0.1.noarch            perl-Error-1:0.17029-5.amzn2023.0.2.noarch
  perl-File-Find-1.37-477.amzn2023.0.7.noarch          perl-Git-2.50.1-1.amzn2023.0.1.noarch
  perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64          perl-lib-0.65-477.amzn2023.0.7.x86_64

Complete!
[ec2-user@ip-172-31-38-214 ~]$ git clone https://github.com/anuranjk/sample-web
Cloning into 'sample-web'...
remote: Enumerating objects: 15, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (12/12), done.
remote: Total 15 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (15/15), 6.24 MiB | 15.70 MiB/s, done.
[ec2-user@ip-172-31-38-214 ~]$ ls
sample-web
[ec2-user@ip-172-31-38-214 ~]$ cd sample web
-bash: cd: too many arguments
[ec2-user@ip-172-31-38-214 ~]$ cd sample-web
[ec2-user@ip-172-31-38-214 sample-web]$ ls
quote-app.rar
[ec2-user@ip-172-31-38-214 sample-web]$ sudo yum install -y epel-release
sudo yum install -y unrar
```

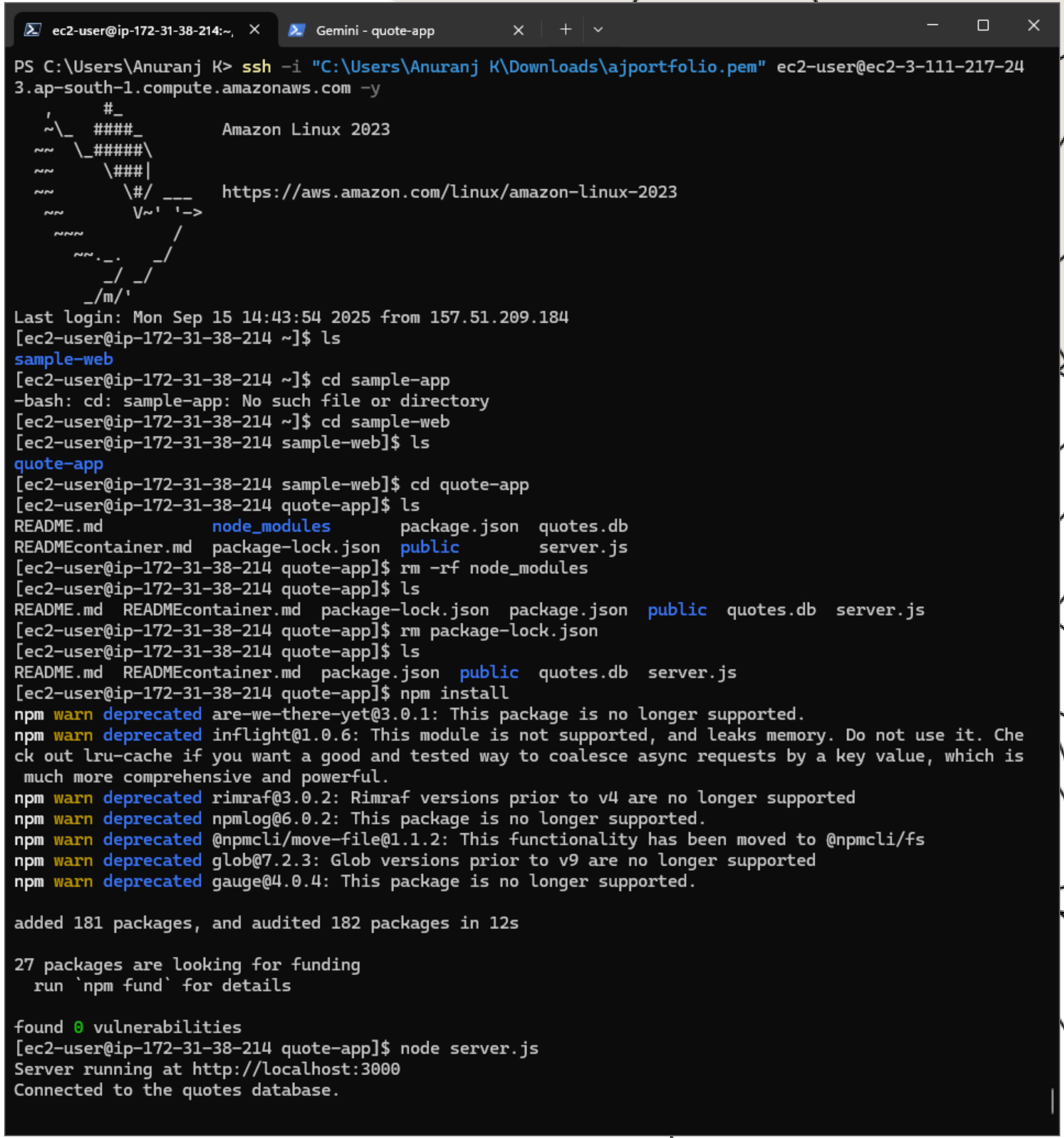


```
  inflating: quote-app/node_modules/util-deprecate/History.md
  inflating: quote-app/node_modules/util-deprecate/LICENSE
  inflating: quote-app/node_modules/util-deprecate/node.js
  inflating: quote-app/node_modules/util-deprecate/package.json
  inflating: quote-app/node_modules/util-deprecate/README.md
   creating: quote-app/node_modules/vary/
  inflating: quote-app/node_modules/vary/HISTORY.md
  inflating: quote-app/node_modules/vary/index.js
  inflating: quote-app/node_modules/vary/LICENSE
  inflating: quote-app/node_modules/vary/package.json
  inflating: quote-app/node_modules/vary/README.md
   creating: quote-app/node_modules/which/
   creating: quote-app/node_modules/which/bin/
  inflating: quote-app/node_modules/which/bin/node-which
  inflating: quote-app/node_modules/which/CHANGELOG.md
  inflating: quote-app/node_modules/which/LICENSE
  inflating: quote-app/node_modules/which/package.json
  inflating: quote-app/node_modules/which/README.md
  inflating: quote-app/node_modules/which/which.js
   creating: quote-app/node_modules/wide-align/
  inflating: quote-app/node_modules/wide-align/align.js
  inflating: quote-app/node_modules/wide-align/LICENSE
  inflating: quote-app/node_modules/wide-align/package.json
  inflating: quote-app/node_modules/wide-align/README.md
   creating: quote-app/node_modules/wrappy/
  inflating: quote-app/node_modules/wrappy/LICENSE
  inflating: quote-app/node_modules/wrappy/package.json
  inflating: quote-app/node_modules/wrappy/README.md
  inflating: quote-app/node_modules/wrappy/wrappy.js
   creating: quote-app/node_modules/yallist/
  inflating: quote-app/node_modules/yallist/iterator.js
  inflating: quote-app/node_modules/yallist/LICENSE
  inflating: quote-app/node_modules/yallist/package.json
  inflating: quote-app/node_modules/yallist/README.md
  inflating: quote-app/node_modules/yallist/yallist.js
  inflating: quote-app/package-lock.json
  inflating: quote-app/package.json
   creating: quote-app/public/
  inflating: quote-app/public/index.html
  inflating: quote-app/public/script.js
  inflating: quote-app/public/styles.css
  inflating: quote-app/quotes.db
  inflating: quote-app/README.md
  inflating: quote-app/READMEcontainer.md
  inflating: quote-app/server.js
[ec2-user@ip-172-31-38-214 sample-web]$ ls
quote-app  quote-app.zip
[ec2-user@ip-172-31-38-214 sample-web]$ rm quote-app.zip
[ec2-user@ip-172-31-38-214 sample-web]$ ls
quote-app
[ec2-user@ip-172-31-38-214 sample-web]$
```

Then I navigated to the app folder and installed necessary npm packages(npm install) then I start the server but it failed because I was using the app which was built on another machine coping node_modules from another os breaks compatibility. So I fixed it by deleting the node_modules folder and packages-lock.json file and tried (npm install) again in the app folder then ran it (node server.js)

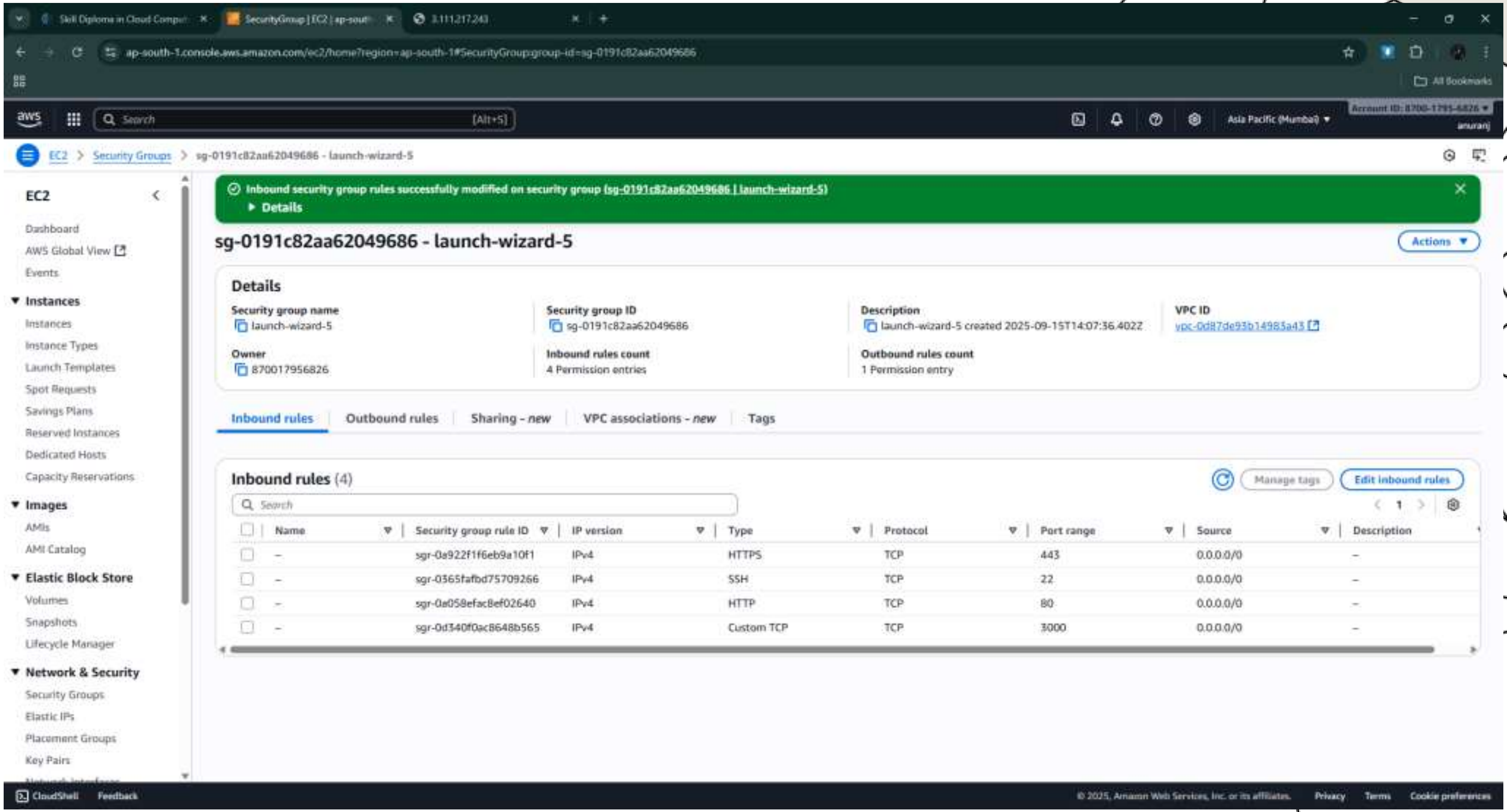


```
This "invalid ELF header" error almost always means that a native Node.js module (in this
case, sqlite3) was compiled for one operating system and architecture (like your Windows
machine) and then you tried to run it on another (Amazon Linux).


The node_modules folder contains compiled code that is specific to the system where npm
install was run. You can't just copy it from a Windows machine to a Linux machine and
expect it to work.


To fix this on your Amazon Linux server, you need to rebuild the sqlite3 module for the
correct environment. Here's what you should do on the server:


 1. Navigate to your project directory: cd /home/ec2-user/sample-web/quote-app
This will download and compile a version of sqlite3 that is compatible with Amazon Linux.
After that, node server.js should work correctly.
```



```
PS C:\Users\Anuranj K> ssh -i "C:\Users\Anuranj K\Downloads\ajportfolio.pem" ec2-user@ec2-3-111-217-24
3.ap-south-1.compute.amazonaws.com -y
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Mon Sep 15 14:43:54 2025 from 157.51.209.184
[ec2-user@ip-172-31-38-214 ~]$ ls
sample-web
[ec2-user@ip-172-31-38-214 ~]$ cd sample-app
-bash: cd: sample-app: No such file or directory
[ec2-user@ip-172-31-38-214 ~]$ cd sample-web
[ec2-user@ip-172-31-38-214 sample-web]$ ls
quote-app
[ec2-user@ip-172-31-38-214 sample-web]$ cd quote-app
[ec2-user@ip-172-31-38-214 quote-app]$ ls
README.md           node_modules       package.json  quotes.db
READMEcontainer.md  package-lock.json  public        server.js
[ec2-user@ip-172-31-38-214 quote-app]$ rm -rf node_modules
[ec2-user@ip-172-31-38-214 quote-app]$ ls
README.md  READMEcontainer.md  package-lock.json  package.json  public  quotes.db  server.js
[ec2-user@ip-172-31-38-214 quote-app]$ rm package-lock.json
[ec2-user@ip-172-31-38-214 quote-app]$ ls
README.md  READMEcontainer.md  package.json  public  quotes.db  server.js
[ec2-user@ip-172-31-38-214 quote-app]$ npm install
npm warn deprecated are-we-there-yet@3.0.1: This package is no longer supported.
npm warn deprecated inflight@1.0.6: This module is not supported, and leaks memory. Do not use it. Che
ck out lru-cache if you want a good and tested way to coalesce async requests by a key value, which is
 much more comprehensive and powerful.
npm warn deprecated rimraf@3.0.2: Rimraf versions prior to v4 are no longer supported
npm warn deprecated npmlog@6.0.2: This package is no longer supported.
npm warn deprecated @npmcli/move-file@1.1.2: This functionality has been moved to @npmcli/fs
npm warn deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
npm warn deprecated gauge@4.0.4: This package is no longer supported.

added 181 packages, and audited 182 packages in 12s

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

found 0 vulnerabilities
[ec2-user@ip-172-31-38-214 quote-app]$ node server.js
Server running at http://localhost:3000
Connected to the quotes database.
```

Then I tried accessing the website using its public ip address but it failed, it was because the node.js use to run the app In **port:3000** and the security group didn't allowed it so I changed it by adding a new inbound rule in sg.



Inbound security group rules successfully modified on security group (sg-0191c82aa62049686 | launch-wizard-5)

sg-0191c82aa62049686 - launch-wizard-5

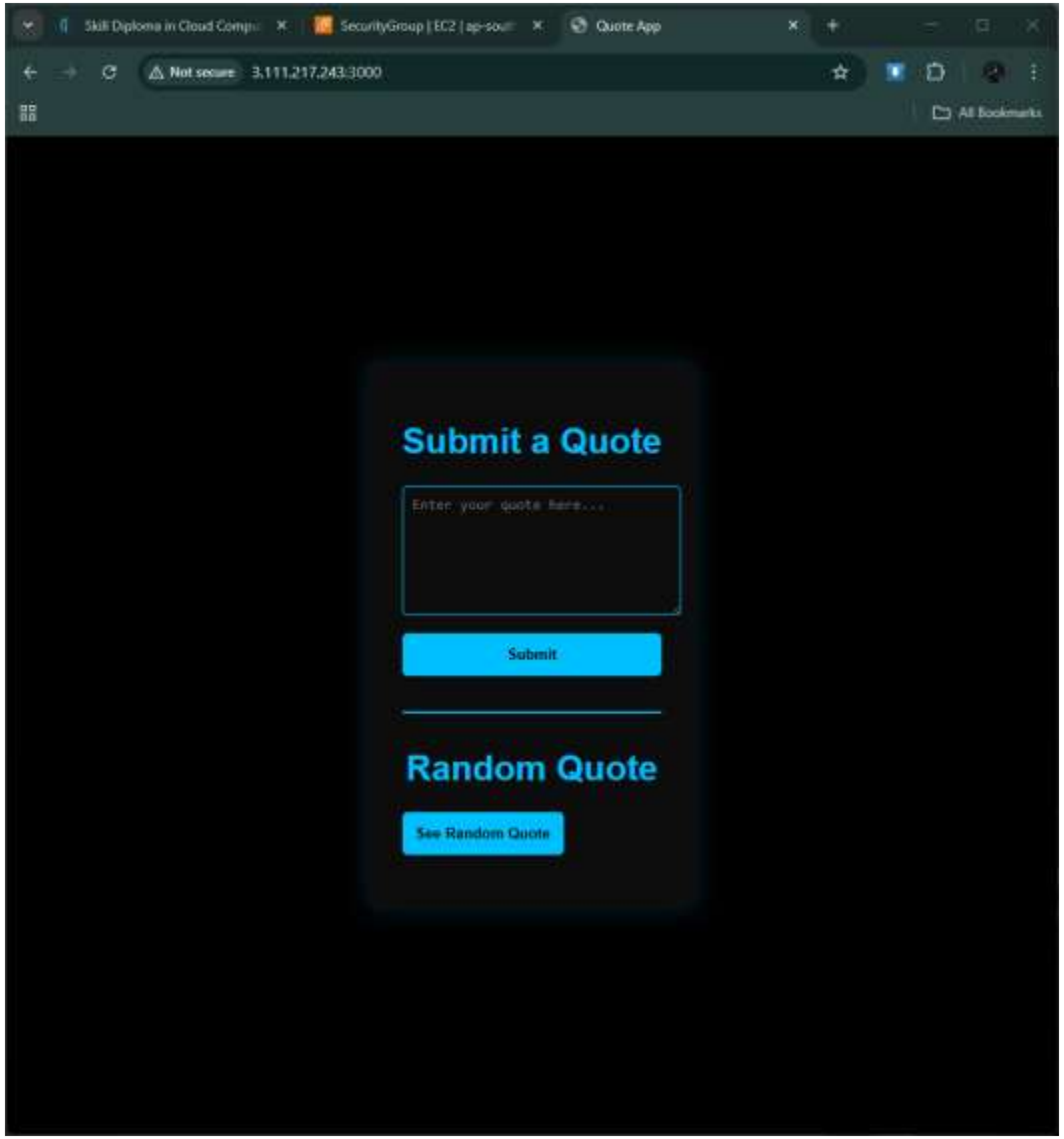
Details

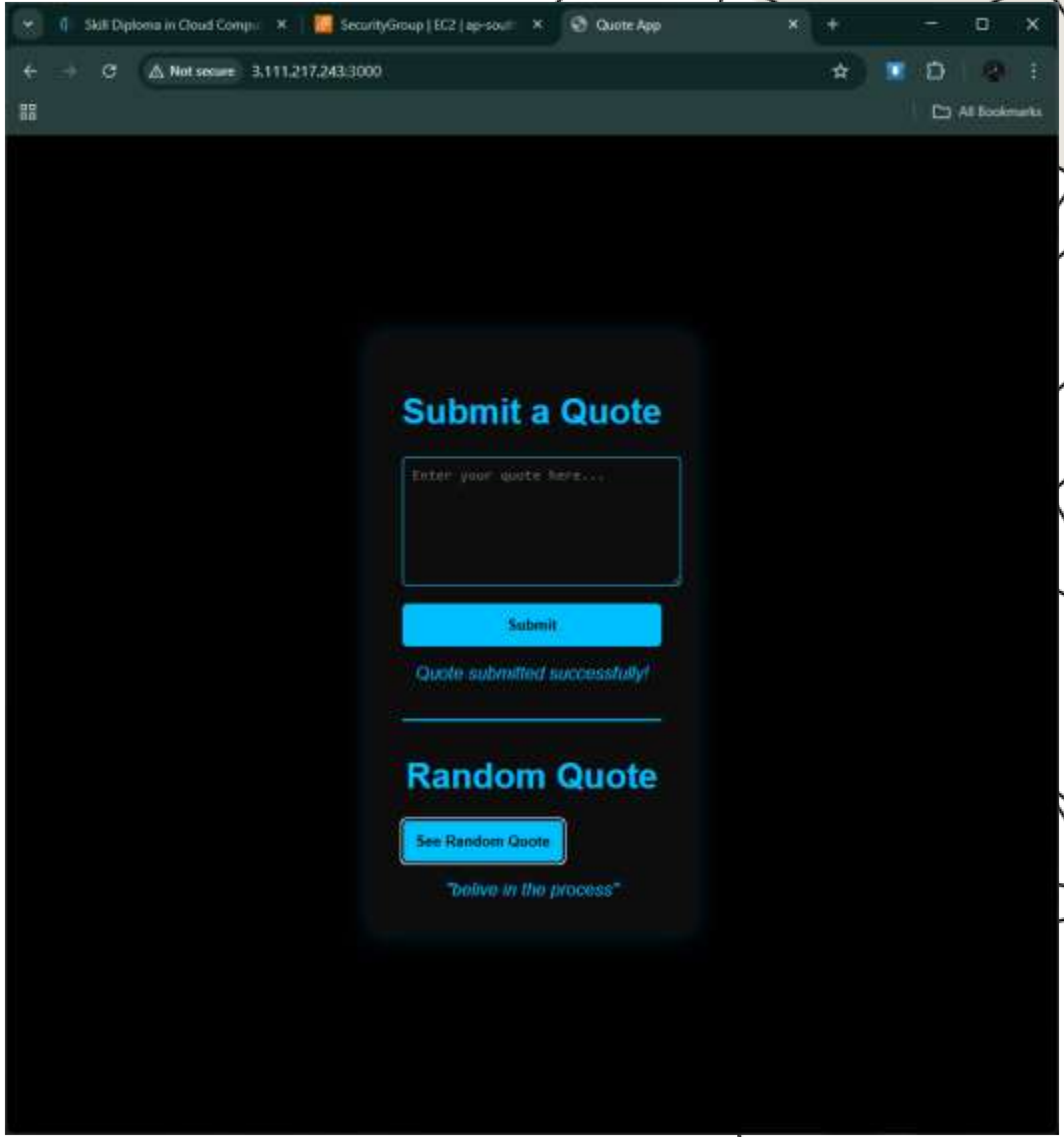
Security group name launch-wizard-5	Security group ID sg-0191c82aa62049686	Description launch-wizard-5 created 2025-09-15T14:07:36.402Z	VPC ID vpc-0d87de93b14983a43
Owner 870017956826	Inbound rules count 4 Permission entries	Outbound rules count 1 Permission entry	

Inbound rules (4)

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source	Description
–	sgr-0a922f1f6eb9a10f1	IPv4	HTTPS	TCP	443	0.0.0.0/0	–
–	sgr-0365fafbd75709266	IPv4	SSH	TCP	22	0.0.0.0/0	–
–	sgr-0a058efac8ef02640	IPv4	HTTP	TCP	80	0.0.0.0/0	–
–	sgr-0d340f0ac8648b565	IPv4	Custom TCP	TCP	3000	0.0.0.0/0	–

Now the website was accessible through port 3000. and I tested it by uploading some quotes and checking random quotes.



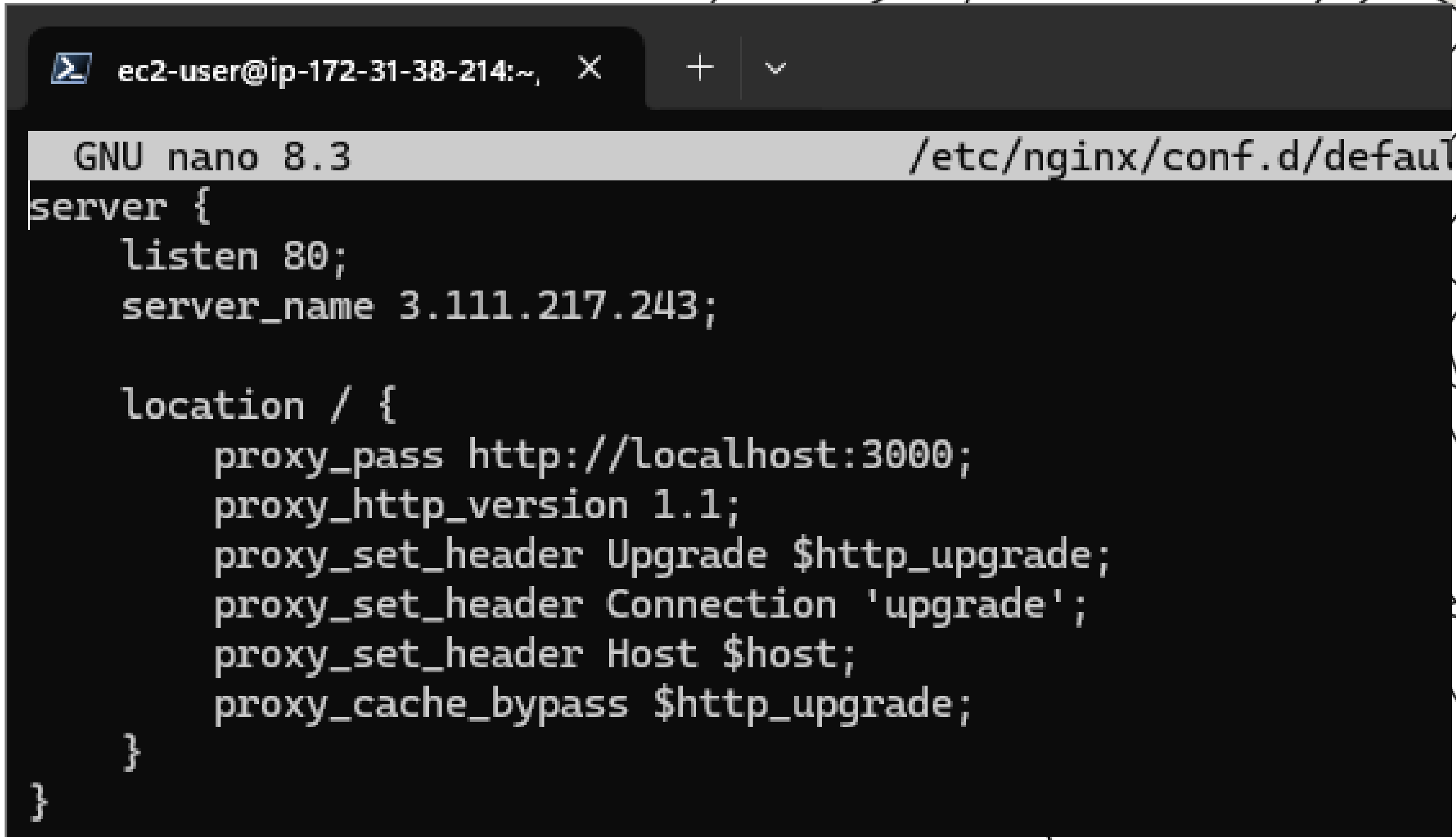


Quote submitted successfully!

"belive in the process"

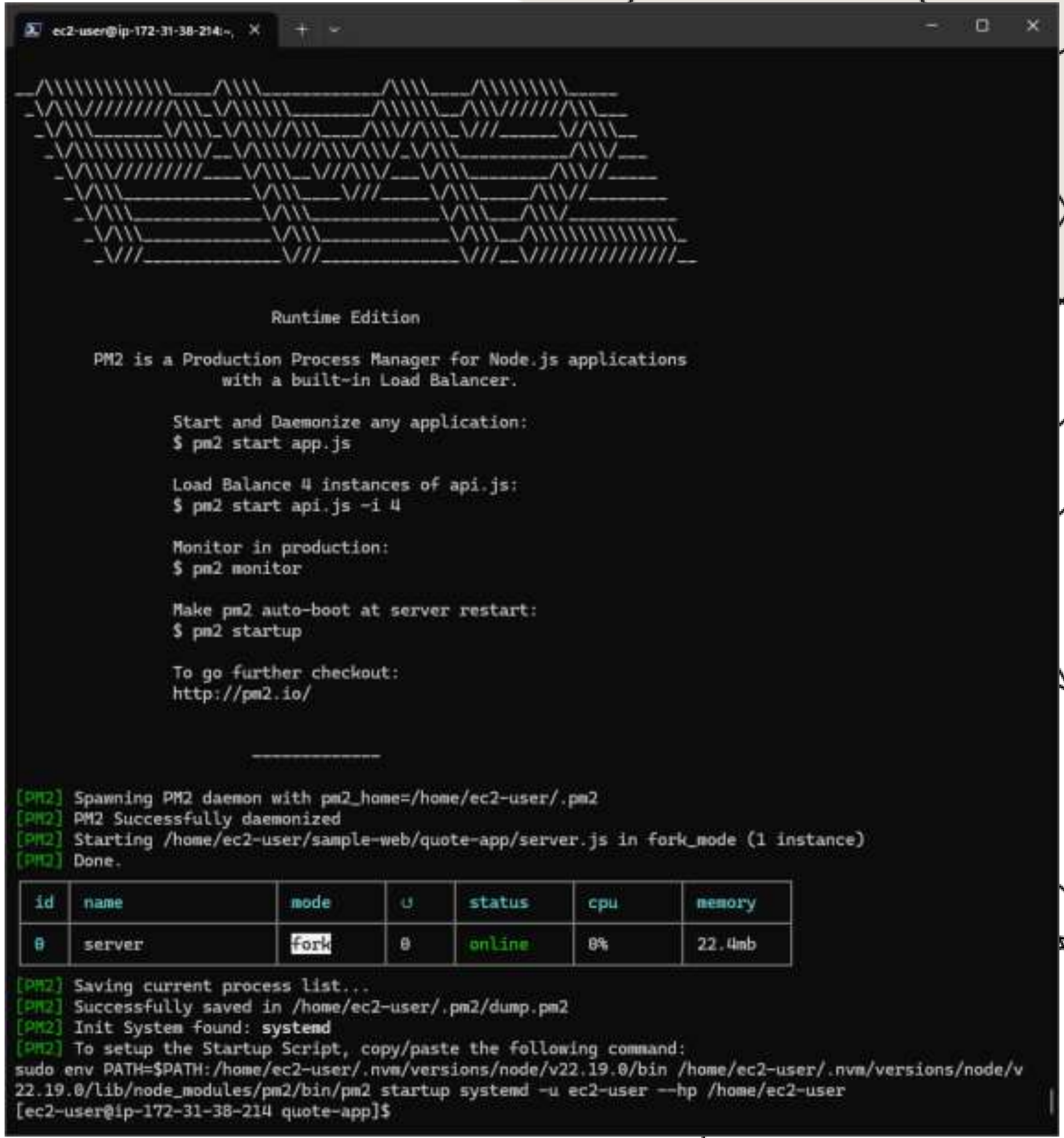
Now it was time to make the website accessible via port 80 so I installed nginx for it (**sudo yum install nginx –y**) and set up reverse proxy by configuring this file(**sudo nano /etc/nginx/conf.d/default.conf**) after saving the config estart nginx **(sudo systemctl restart ngnix**)

```
server {
    listen 80;
    server_name yourdomain.com
www.yourdomain.com;

    location / {
        proxy_pass
http://localhost:3000;
        proxy_http_version 1.1;
        proxy_set_header Upgrade
$http_upgrade;
        proxy_set_header Connection
'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass
$http_upgrade;
    }
}
```



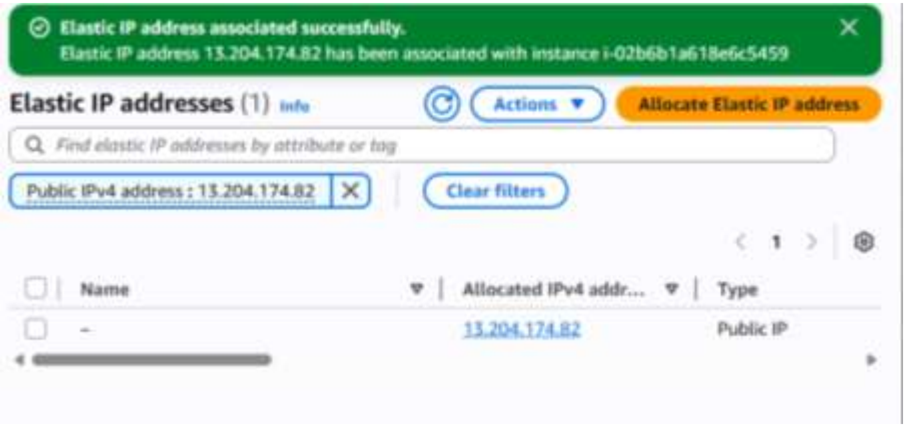
```
  GNU nano 8.3                          /etc/nginx/conf.d/defaul
server {
    listen 80;
    server_name 3.111.217.243;

    location / {
        proxy_pass http://localhost:3000;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}
```

Then I installed process manager so that it can run efficiently and restart if fail. (**npm install pm2**)



```
__/\\\\\\\\\\\\\____/\\\\____________/\\\\____/\\\\\\\\\_____
 _\/\\\/////////\\\_\/\\\\\\________/\\\\\\__/\\\///////\\\___
  _\/\\\_______\/\\\_\/\\\//\\\____/\\\//\\\_\///______\//\\\__
   _\/\\\\\\\\\\\\\/__\/\\\\///\\\/\\\/_\/\\\___________/\\\/___
    _\/\\\/////////____\/\\\__\///\\\/___\/\\\________/\\\//_____
     _\/\\\_____________\/\\\____\///_____\/\\\_____/\\\//________
      _\/\\\_____________\/\\\_____________\/\\\___/\\\/___________
       _\/\\\_____________\/\\\_____________\/\\\__/\\\\\\\\\\\\\\\_
        _\///______________\///______________\///__\///////////////__


                          Runtime Edition

        PM2 is a Production Process Manager for Node.js applications
                     with a built-in Load Balancer.

                Start and Daemonize any application:
                $ pm2 start app.js

                Load Balance 4 instances of api.js:
                $ pm2 start api.js -i 4

                Monitor in production:
                $ pm2 monitor

                Make pm2 auto-boot at server restart:
                $ pm2 startup

                To go further checkout:
                http://pm2.io/


                        -------------

[PM2] Spawning PM2 daemon with pm2_home=/home/ec2-user/.pm2
[PM2] PM2 Successfully daemonized
[PM2] Starting /home/ec2-user/sample-web/quote-app/server.js in fork_mode (1 instance)
[PM2] Done.
┌────┬───────────┬──────────┬──────┬───────────┬──────────┬──────────┐
│ id │ name      │ mode     │ ↺    │ status    │ cpu      │ memory   │
├────┼───────────┼──────────┼──────┼───────────┼──────────┼──────────┤
│ 0  │ server    │ fork     │ 0    │ online    │ 0%       │ 22.4mb   │
└────┴───────────┴──────────┴──────┴───────────┴──────────┴──────────┘
[PM2] Saving current process list...
[PM2] Successfully saved in /home/ec2-user/.pm2/dump.pm2
[PM2] Init System found: systemd
[PM2] To setup the Startup Script, copy/paste the following command:
sudo env PATH=$PATH:/home/ec2-user/.nvm/versions/node/v22.19.0/bin /home/ec2-user/.nvm/versions/node/v
22.19.0/lib/node_modules/pm2/bin/pm2 startup systemd -u ec2-user --hp /home/ec2-user
[ec2-user@ip-172-31-38-214 quote-app]$
```

Then I wanted to add a domain name for the website. First of all I allocated a elastip ip for my server then I changed the nginx config so that my new ip will work. Then I navigated to my already purchased domain and added an **A record .** Then the nginx config was again changed instead of giving ip I entered the domain name there.



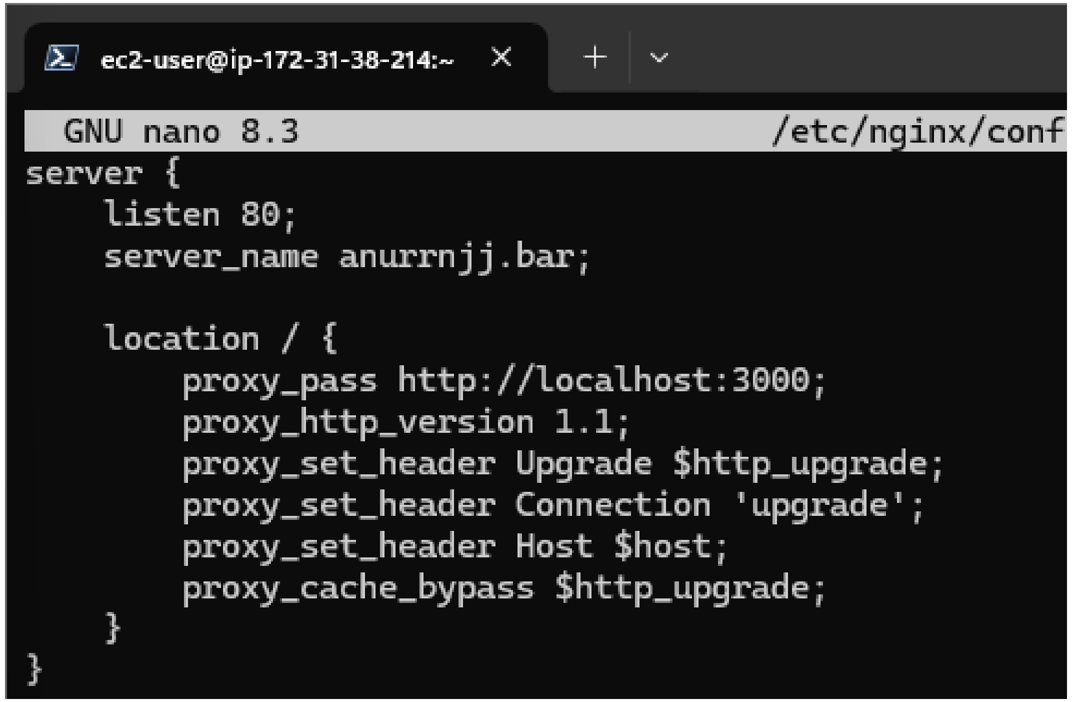
Elastic IP address associated successfully.
Elastic IP address 13.204.174.82 has been associated with instance i-02b6b1a618e6c5459

Elastic IP addresses (1)

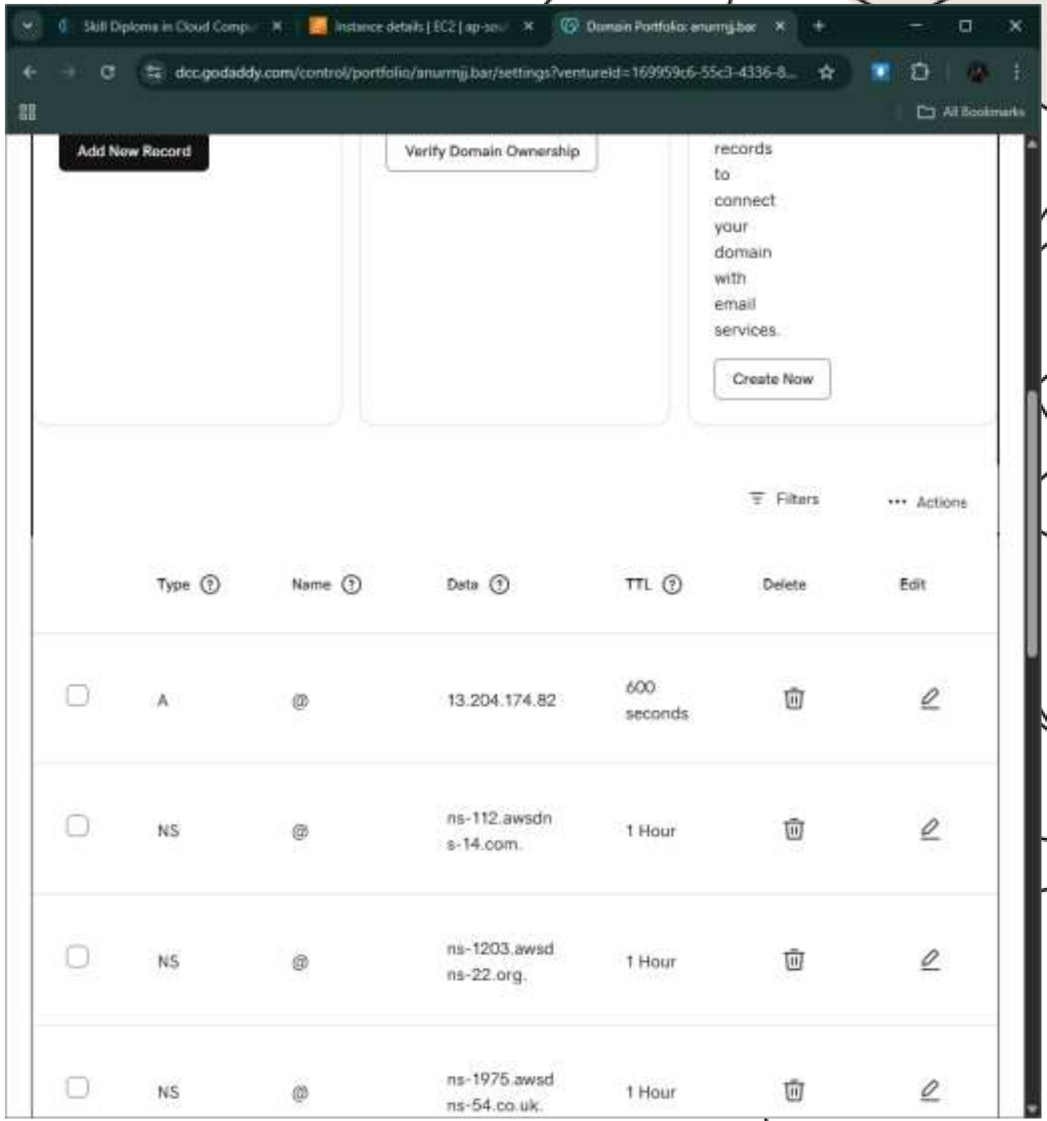
Info
Actions ▼
Allocate Elastic IP address
Find elastic IP addresses by attribute or tag
Clear filters
1

Public IPv4 address : 13.204.174.82

□	Name	Allocated IPv4 addr...	Type
□	-	13.204.174.82	Public IP



```
 GNU nano 8.3                     /etc/nginx/conf
server {
    listen 80;
    server_name anurrnjj.bar;

    location / {
        proxy_pass http://localhost:3000;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}
```

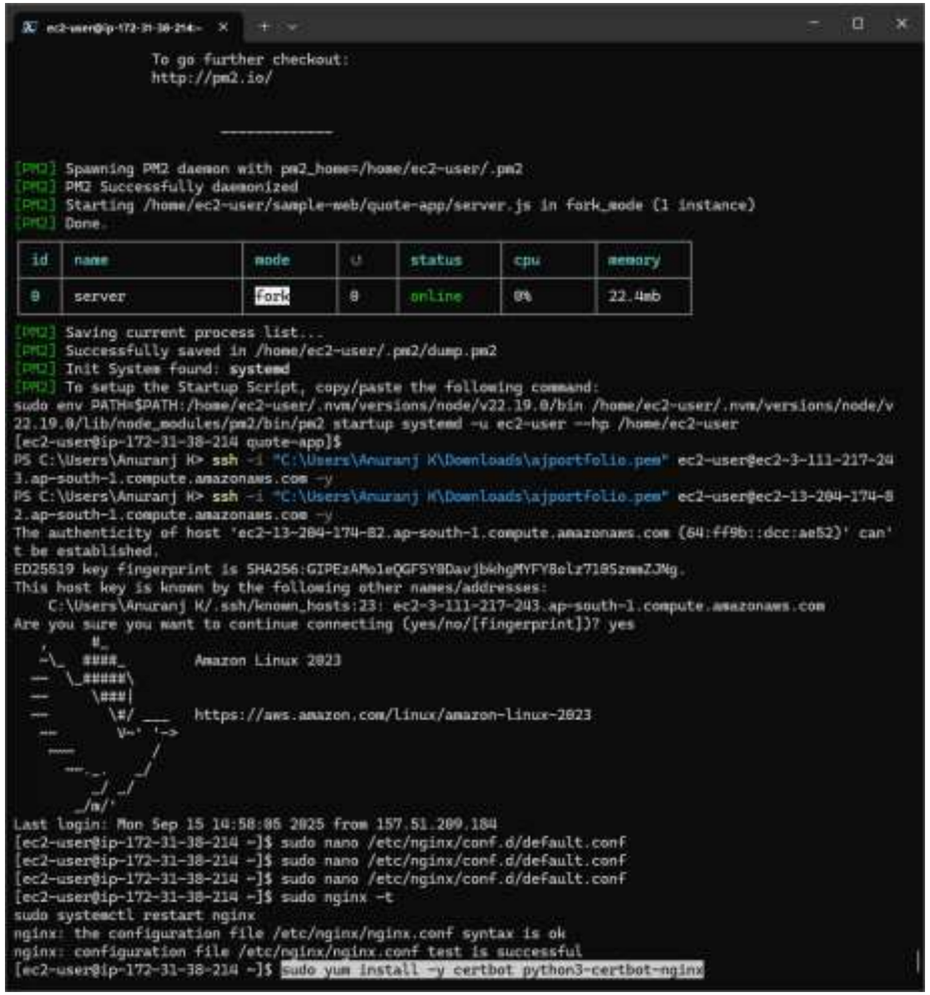


Skill Diploma in Cloud Comp
Instance details | EC2 | ap-sou
Domain Portfolio: anurrnjj.bar
dcc.godaddy.com/control/portfolio/anurrnjj.bar/settings?ventureId=169959c6-55c3-4336-8...
All Bookmarks
Add New Record
Verify Domain Ownership
Create Now
Filters
Actions

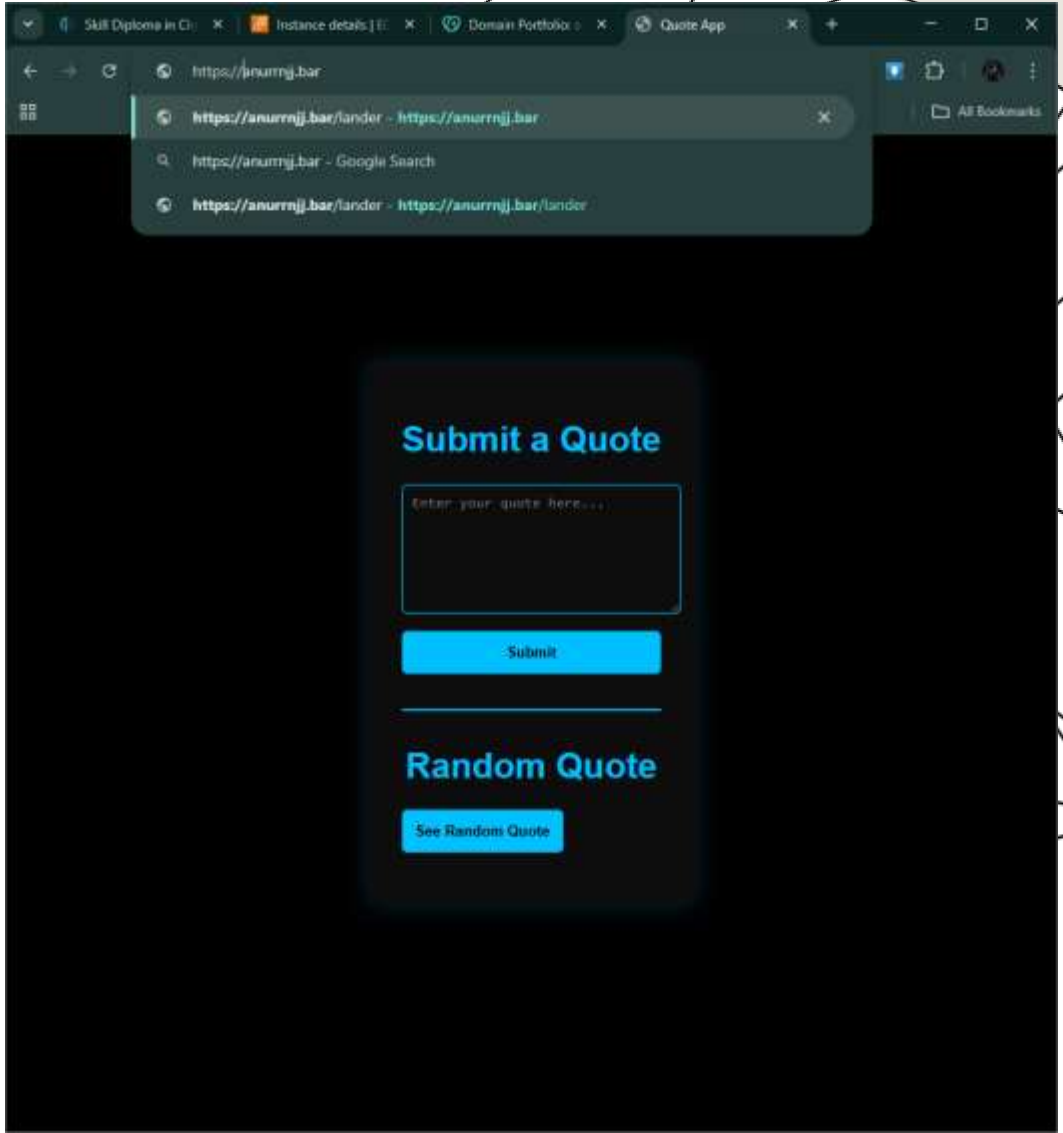
records to connect your domain with email services.

	Type	Name	Data	TTL	Delete	Edit
□	A	@	13.204.174.82	600 seconds		
□	NS	@	ns-112.awsdns-14.com.	1 Hour		
□	NS	@	ns-1203.awsdns-22.org.	1 Hour		
□	NS	@	ns-1975.awsdns-54.co.uk.	1 Hour		

To get access via https I need to assign it to a ssl certificate so I continued by installing certbort wich was free (**sudo yum install –y certbot python3-certbot-ngnix**). Then I ran the certbot to obtain and install the certificate. (**sudo certbot --nginx -d yourdomain.com**)



```
                To go further checkout:
                http://pm2.io/


                        --------------

[PM2] Spawning PM2 daemon with pm2_home=/home/ec2-user/.pm2
[PM2] PM2 Successfully daemonized
[PM2] Starting /home/ec2-user/sample-web/quote-app/server.js in fork_mode (1 instance)
[PM2] Done.
┌────┬────────────┬──────────┬────┬──────────┬──────────┬──────────┐
│ id │ name       │ mode     │ ↺  │ status   │ cpu      │ memory   │
├────┼────────────┼──────────┼────┼──────────┼──────────┼──────────┤
│ 0  │ server     │ fork     │ 0  │ online   │ 0%       │ 22.4mb   │
└────┴────────────┴──────────┴────┴──────────┴──────────┴──────────┘
[PM2] Saving current process list...
[PM2] Successfully saved in /home/ec2-user/.pm2/dump.pm2
[PM2] Init System found: systemd
[PM2] To setup the Startup Script, copy/paste the following command:
sudo env PATH=$PATH:/home/ec2-user/.nvm/versions/node/v22.19.0/bin /home/ec2-user/.nvm/versions/node/v
22.19.0/lib/node_modules/pm2/bin/pm2 startup systemd -u ec2-user --hp /home/ec2-user
[ec2-user@ip-172-31-38-214 quote-app]$
PS C:\Users\Anuranj K> ssh -i "C:\Users\Anuranj K\Downloads\ajportfolio.pem" ec2-user@ec2-3-111-217-24
3.ap-south-1.compute.amazonaws.com -y
PS C:\Users\Anuranj K> ssh -i "C:\Users\Anuranj K\Downloads\ajportfolio.pem" ec2-user@ec2-13-204-174-8
2.ap-south-1.compute.amazonaws.com -y
The authenticity of host 'ec2-13-204-174-82.ap-south-1.compute.amazonaws.com (64:ff9b::dcc:ae52)' can'
t be established.
ED25519 key fingerprint is SHA256:GIPEzAMoleQGFSY8DavjbkhgMYFY8olz7l0SzmmZJNg.
This host key is known by the following other names/addresses:
    C:\Users\Anuranj K/.ssh/known_hosts:23: ec2-3-111-217-243.ap-south-1.compute.amazonaws.com
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Mon Sep 15 14:58:05 2025 from 157.51.209.184
[ec2-user@ip-172-31-38-214 ~]$ sudo nano /etc/nginx/conf.d/default.conf
[ec2-user@ip-172-31-38-214 ~]$ sudo nano /etc/nginx/conf.d/default.conf
[ec2-user@ip-172-31-38-214 ~]$ sudo nano /etc/nginx/conf.d/default.conf
[ec2-user@ip-172-31-38-214 ~]$ sudo nginx -t
sudo systemctl restart nginx
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
[ec2-user@ip-172-31-38-214 ~]$ sudo yum install -y certbot python3-certbot-nginx
```


Skill Diploma in Cl
Instance details | E
Domain Portfolio
Quote App
https://anurnj.bar
https://anurnj.bar/lander - https://anurnj.bar
https://anurnj.bar - Google Search
https://anurnj.bar/lander - https://anurnj.bar/lander
All Bookmarks
Submit a Quote
Enter your quote here...
Submit
Random Quote
See Random Quote