



# Introduction To Cloud Computing

## DV1566

### Project Report

#### **Group 20**

Sree Lakshmi Hiranmayee Kadali

Avan Chowdary Gogineni

Lahari Gaddam

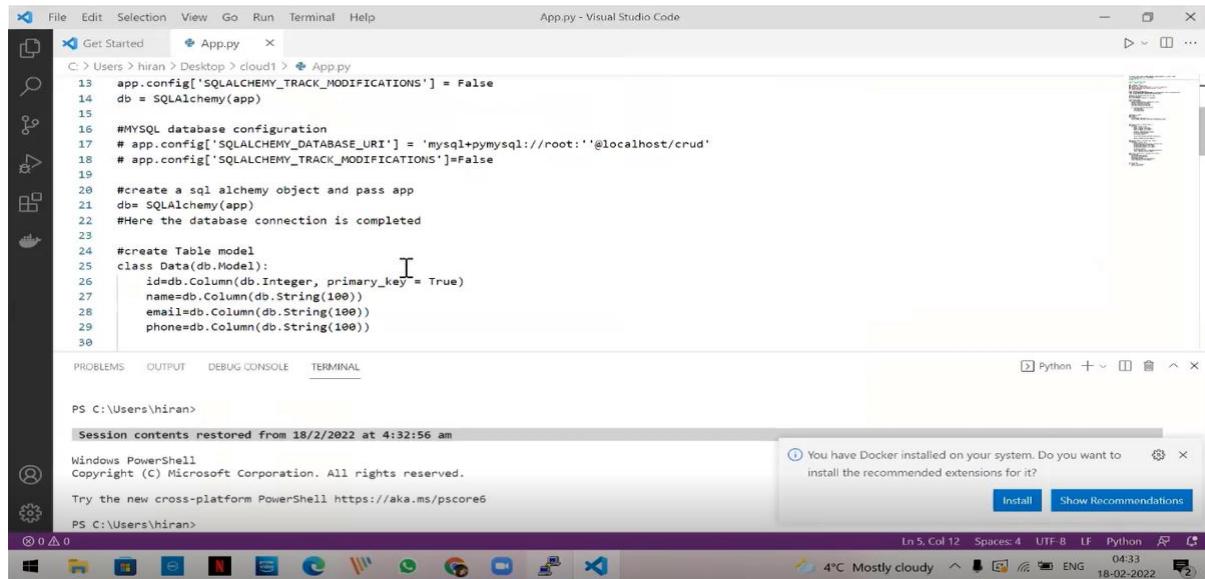
## Problem statement and project description:

This project is to create and deploy a CRUD application that can create, remove, update, delete, records of the employee and give a user-friendly experience using flask framework and to create a VPC and launch the instance on EC2 for running applications on the Amazon Web Services (AWS) infrastructure. As the Application would be tested against increased concurrent requests from users, based on CPU utilization we perform auto-scaling and provide high availability of computation.

## Creating a flask application:

We have written a code for the application using various script files like html, CSS, and jQuery. We then run the application by running app.py. We have created a MySQL database, and this is our database model.

Finally, running the complete project will give out the following result.



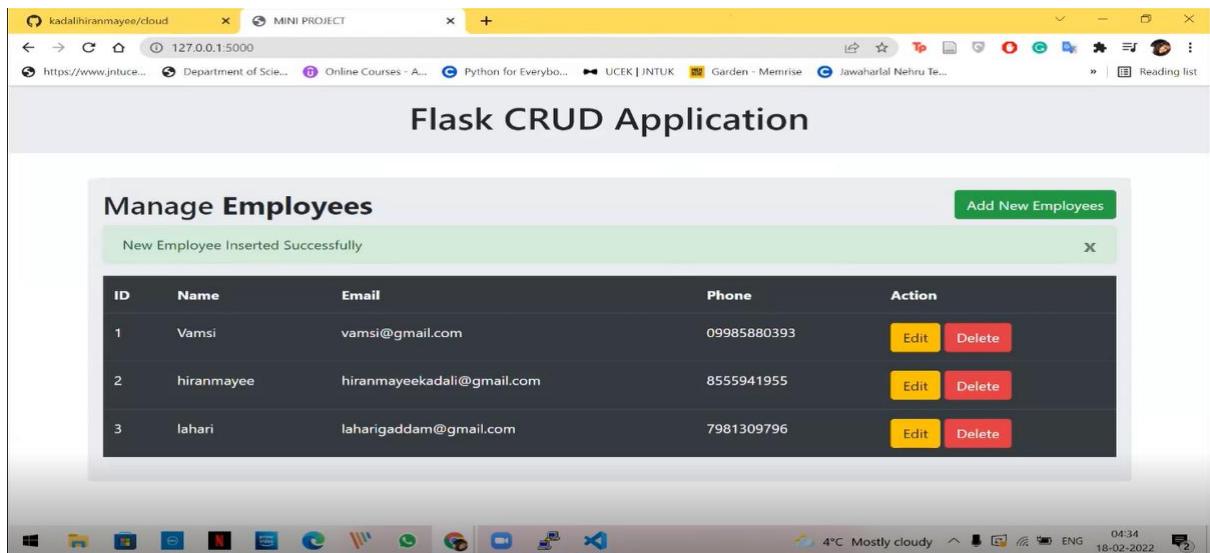
```
C:\> Users > hiran > Desktop > cloud1 > App.py
 13 app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
 14 db = SQLAlchemy(app)
 15
 16 #MySQL database configuration
 17 # app.config['SQLALCHEMY_DATABASE_URI'] = 'mysql+pymysql://root:' '@localhost/crud'
 18 # app.config['SQLALCHEMY_TRACK_MODIFICATIONS']=False
 19
 20 #create a sql alchemy object and pass app
 21 db= SQLAlchemy(app)
 22 #Here the database connection is completed
 23
 24 #create Table model
 25 class Data(db.Model):
 26     id=db.Column(db.Integer, primary_key=True)
 27     name=db.Column(db.String(100))
 28     email=db.Column(db.String(100))
 29     phone=db.Column(db.String(100))
 30
 31
 32
 33
 34
 35
 36
 37
 38
 39
 40
 41
 42
 43
 44
 45
 46
 47
 48
 49
 50
 51
 52
 53
 54
 55
 56
 57
 58
 59
 60
 61
 62
 63
 64
 65
 66
 67
 68
 69
 70
 71
 72
 73
 74
 75
 76
 77
 78
 79
 80
 81
 82
 83
 84
 85
 86
 87
 88
 89
 90
 91
 92
 93
 94
 95
 96
 97
 98
 99
 100
 101
 102
 103
 104
 105
 106
 107
 108
 109
 110
 111
 112
 113
 114
 115
 116
 117
 118
 119
 120
 121
 122
 123
 124
 125
 126
 127
 128
 129
 130
 131
 132
 133
 134
 135
 136
 137
 138
 139
 140
 141
 142
 143
 144
 145
 146
 147
 148
 149
 150
 151
 152
 153
 154
 155
 156
 157
 158
 159
 160
 161
 162
 163
 164
 165
 166
 167
 168
 169
 170
 171
 172
 173
 174
 175
 176
 177
 178
 179
 180
 181
 182
 183
 184
 185
 186
 187
 188
 189
 190
 191
 192
 193
 194
 195
 196
 197
 198
 199
 200
 201
 202
 203
 204
 205
 206
 207
 208
 209
 210
 211
 212
 213
 214
 215
 216
 217
 218
 219
 220
 221
 222
 223
 224
 225
 226
 227
 228
 229
 230
 231
 232
 233
 234
 235
 236
 237
 238
 239
 240
 241
 242
 243
 244
 245
 246
 247
 248
 249
 250
 251
 252
 253
 254
 255
 256
 257
 258
 259
 260
 261
 262
 263
 264
 265
 266
 267
 268
 269
 270
 271
 272
 273
 274
 275
 276
 277
 278
 279
 280
 281
 282
 283
 284
 285
 286
 287
 288
 289
 290
 291
 292
 293
 294
 295
 296
 297
 298
 299
 300
 301
 302
 303
 304
 305
 306
 307
 308
 309
 310
 311
 312
 313
 314
 315
 316
 317
 318
 319
 320
 321
 322
 323
 324
 325
 326
 327
 328
 329
 330
 331
 332
 333
 334
 335
 336
 337
 338
 339
 340
 341
 342
 343
 344
 345
 346
 347
 348
 349
 350
 351
 352
 353
 354
 355
 356
 357
 358
 359
 360
 361
 362
 363
 364
 365
 366
 367
 368
 369
 370
 371
 372
 373
 374
 375
 376
 377
 378
 379
 380
 381
 382
 383
 384
 385
 386
 387
 388
 389
 390
 391
 392
 393
 394
 395
 396
 397
 398
 399
 400
 401
 402
 403
 404
 405
 406
 407
 408
 409
 410
 411
 412
 413
 414
 415
 416
 417
 418
 419
 420
 421
 422
 423
 424
 425
 426
 427
 428
 429
 430
 431
 432
 433
 434
 435
 436
 437
 438
 439
 440
 441
 442
 443
 444
 445
 446
 447
 448
 449
 450
 451
 452
 453
 454
 455
 456
 457
 458
 459
 460
 461
 462
 463
 464
 465
 466
 467
 468
 469
 470
 471
 472
 473
 474
 475
 476
 477
 478
 479
 480
 481
 482
 483
 484
 485
 486
 487
 488
 489
 490
 491
 492
 493
 494
 495
 496
 497
 498
 499
 500
 501
 502
 503
 504
 505
 506
 507
 508
 509
 510
 511
 512
 513
 514
 515
 516
 517
 518
 519
 520
 521
 522
 523
 524
 525
 526
 527
 528
 529
 530
 531
 532
 533
 534
 535
 536
 537
 538
 539
 540
 541
 542
 543
 544
 545
 546
 547
 548
 549
 550
 551
 552
 553
 554
 555
 556
 557
 558
 559
 560
 561
 562
 563
 564
 565
 566
 567
 568
 569
 570
 571
 572
 573
 574
 575
 576
 577
 578
 579
 580
 581
 582
 583
 584
 585
 586
 587
 588
 589
 589
 590
 591
 592
 593
 594
 595
 596
 597
 598
 599
 600
 601
 602
 603
 604
 605
 606
 607
 608
 609
 610
 611
 612
 613
 614
 615
 616
 617
 618
 619
 620
 621
 622
 623
 624
 625
 626
 627
 628
 629
 630
 631
 632
 633
 634
 635
 636
 637
 638
 639
 640
 641
 642
 643
 644
 645
 646
 647
 648
 649
 650
 651
 652
 653
 654
 655
 656
 657
 658
 659
 660
 661
 662
 663
 664
 665
 666
 667
 668
 669
 670
 671
 672
 673
 674
 675
 676
 677
 678
 679
 680
 681
 682
 683
 684
 685
 686
 687
 688
 689
 690
 691
 692
 693
 694
 695
 696
 697
 698
 699
 700
 701
 702
 703
 704
 705
 706
 707
 708
 709
 710
 711
 712
 713
 714
 715
 716
 717
 718
 719
 720
 721
 722
 723
 724
 725
 726
 727
 728
 729
 730
 731
 732
 733
 734
 735
 736
 737
 738
 739
 739
 740
 741
 742
 743
 744
 745
 746
 747
 748
 749
 749
 750
 751
 752
 753
 754
 755
 756
 757
 758
 759
 759
 760
 761
 762
 763
 764
 765
 766
 767
 768
 769
 769
 770
 771
 772
 773
 774
 775
 775
```

The screenshot shows the Visual Studio Code interface with the terminal tab active. The command `app.run(debug=True)` is run, and the output shows the Flask development server starting up on port 5000. It includes a warning about using it in production and provides a debugger PIN.

```
File Edit Selection View Go Run Terminal Help
App.py - Visual Studio Code
Get Started App.py X
C: > Users > hiran > Desktop > cloud1 > App.py > ...
85 app.run(debug=True)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS C:\Users\hiran> & C:/Users/hiran/AppData/Local/Programs/Python/Python310/python.exe c:/Users/hiran/Desktop/cloud1/App.py
* Serving Flask app 'App' (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: on
* Restarting with stat
* Debugger is active!
* Debugger PIN: 117-716-467
* Running on http://127.0.0.1:5000/ [Press CTRL+C to quit)
```

## OUTPUT:



## Virtual private cloud (VPC):

We have created a VPC with the name group\_20 with the required subnets, route tables, internet gateway, security groups, and network ACLs.

**Your VPCs (1/2) Info**

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR
group_20	vpc-0894f2ac91c43aeda	Available	10.0.0.0/20	-

**VPC ID**  
vpc-0894f2ac91c43aeda  
**State** Available  
**Tenancy** Default  
**Default VPC** No  
**Route 53 Resolver DNS Firewall rule groups** -  
**Owner ID** 772707908423

**DNS hostnames** Disabled  
**Main route table** rtb-09281914bbf6d5909 / public  
**Main network ACL** acl-0b8c7b64d1e76a979 / group\_20\_ACL  
**IPv6 pool** -  
**IPv6 CIDR** -

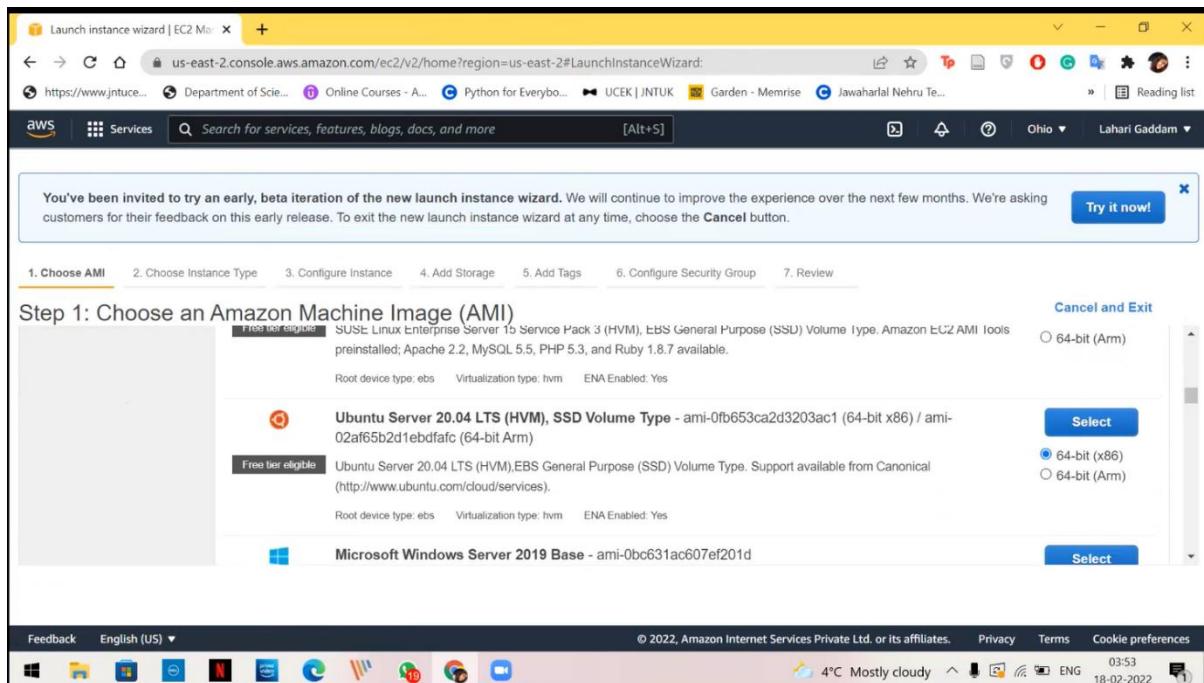
We then create an elastic IP address to the VPC to enable communication with the internet if the VPC does not have a public IPv4 address.

**Summary**

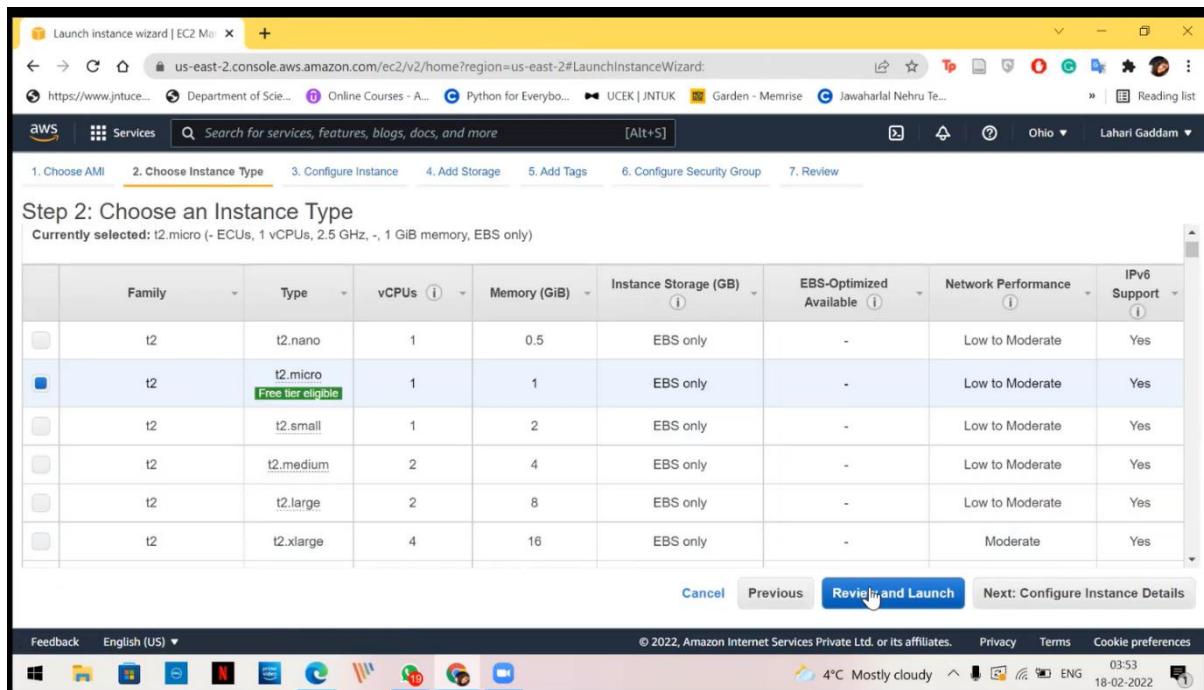
Allocated IPv4 address 3.19.91.242	Type Public IP	Allocation ID eipalloc-01c8da1a4a853eac4	Reverse DNS record -
Association ID eipassoc-08e5973a593b43a3f	Scope VPC	Associated instance ID i-04ff2e84f02d28a6f	Private IP address 10.0.1.61
Network interface ID eni-044b817de8d267977	Network interface owner account ID 772707908423	Public DNS -	NAT Gateway ID -
Address pool Amazon			

## Launching the instance in EC2:

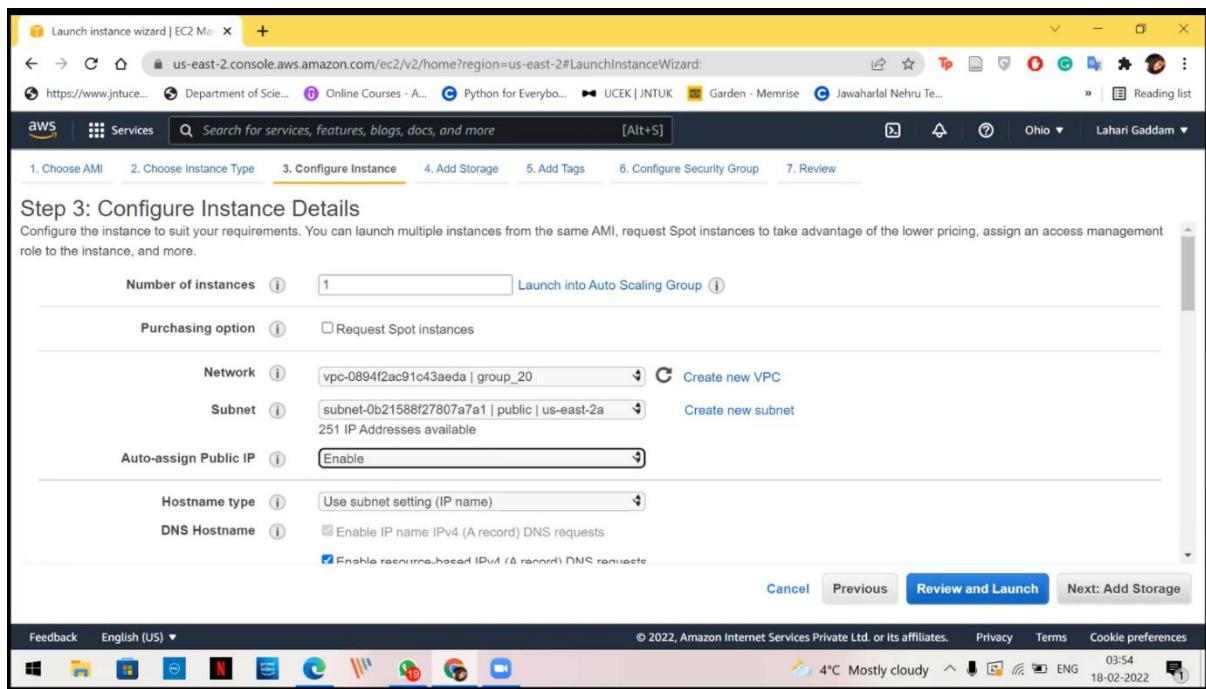
- We have chosen the free tier version of ubuntu server.



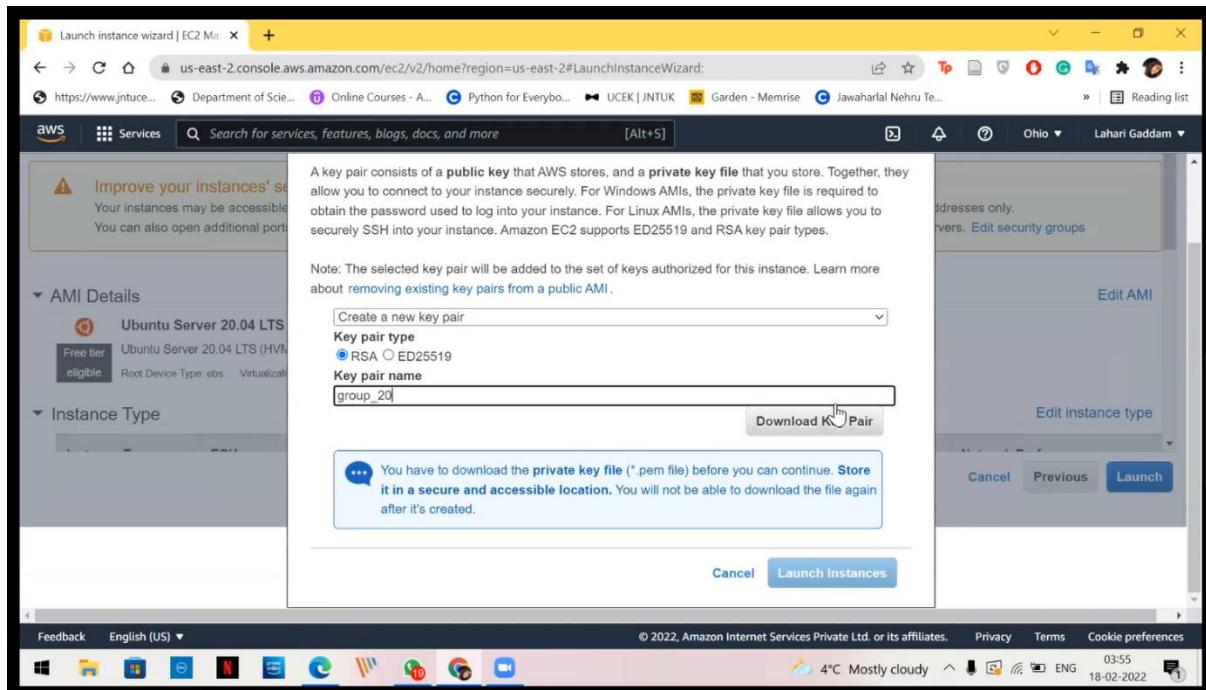
- Choosing the free tier instance type t2.micro



- Configuring the instance details by selecting the VPC that we created, changing the subnet to public and enabling the auto-assign public IP.



- Downloading the key pair for launching the instance in PuTTY.



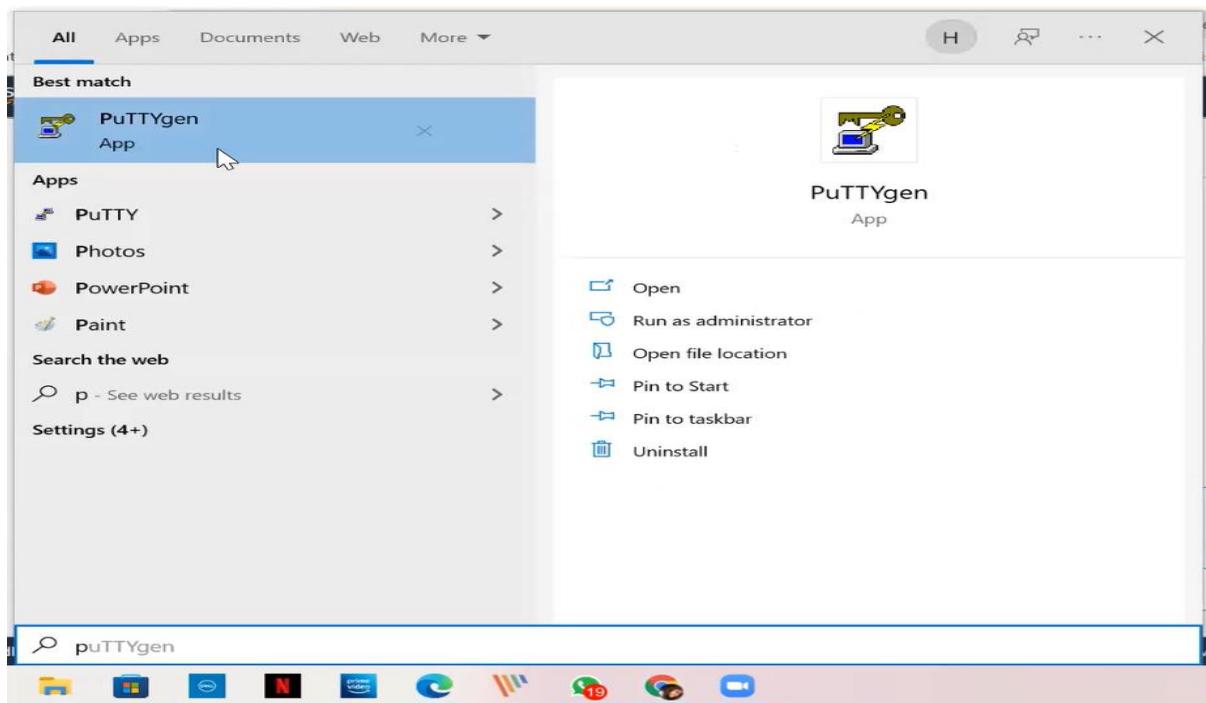
- Name the instance as ubuntu Linux and launch the instance.

The screenshot shows the AWS EC2 Management Console. On the left, there's a sidebar with options like EC2 Dashboard, EC2 Global View, Events, Tags, Limits, and Instances. Under Instances, there are links for Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, and Capacity Reservations. The main area displays a table of instances. One instance, named "ubuntu linux", is highlighted. Its status is "Pending", it's a "t2.micro" type, and it has no alarms. Below the table, a modal window titled "Instance: i-04ff2e84f02d28a6f" is open, showing the "Details" tab. It provides specific information about the instance: Instance ID (i-04ff2e84f02d28a6f), Public IPv4 address (3.145.108.173), and Private IPv4 addresses (10.0.1.61). Other tabs in the modal include Security, Networking, Storage, Status checks, Monitoring, and Tags.

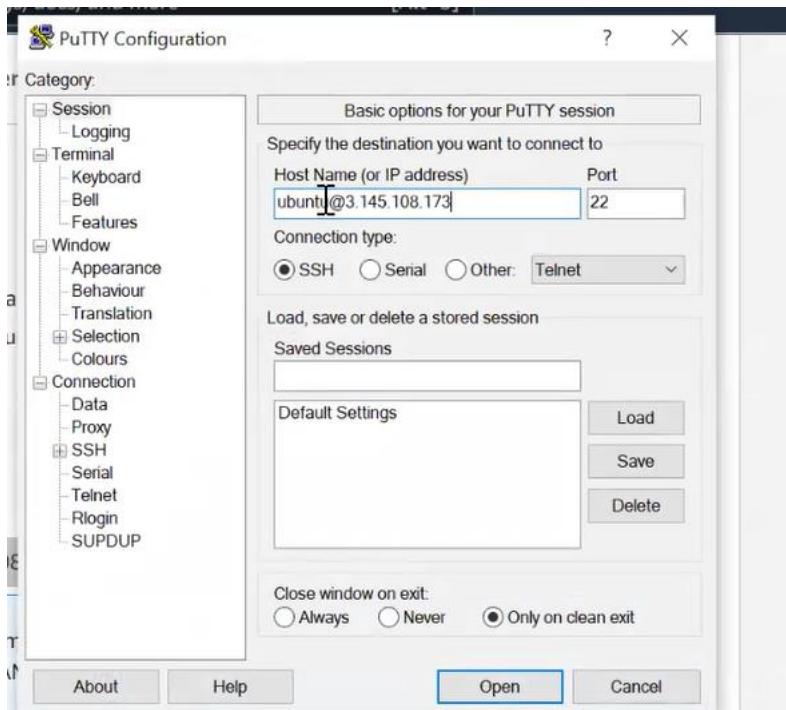
- Copy the public IP of the instance for launching the instance in a SSH client.

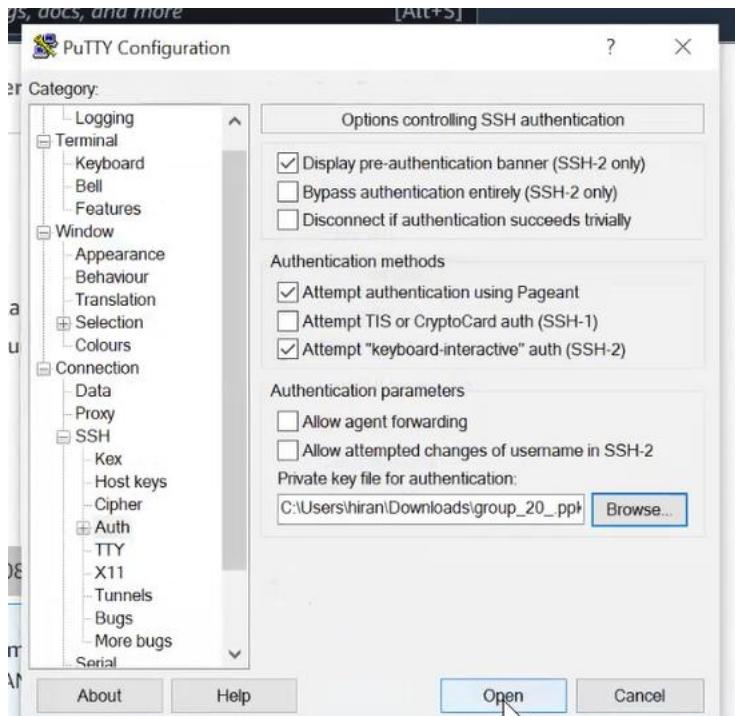
The screenshot shows the "EC2 Instance Connect" section of the AWS EC2 Management Console. It features three tabs: "EC2 Instance Connect", "Session Manager", and "SSH client". The "SSH client" tab is active. It contains instructions for connecting to the instance via SSH. The first step is to "Open an SSH client". The second step is to "Locate your private key file. The key used to launch this instance is group\_20.pem". The third step is to "Run this command, if necessary, to ensure your key is not publicly viewable." Below these steps, there's an example command: "ssh -i 'group\_20.pem' ubuntu@3.145.108.173". At the bottom of the panel, there's a note: "Note: In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name." The browser address bar shows the URL: https://us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#ConnectToInstance:instanceId=i-04ff2e84f02d28a6f.

- Convert the key pair that is in .pem type to .ppk type



- Copy the IP address and browse the .ppk private key.





- Click Open to launch the instance in PuTTY.

```
ubuntu@ip-10-0-1-61: ~
System load: 0.33      Processes:          102
Usage of /: 18.2% of 7.69GB   Users logged in:    0
Memory usage: 20%           IPv4 address for eth0: 10.0.1.61
Swap usage:  0%
1 update can be applied immediately.
To see these additional updates run: apt list --upgradable

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

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-10-0-1-61:~$
```

## DEPLOYMENT OF CRUD WEBSITE USING EC2 INSTANCE:

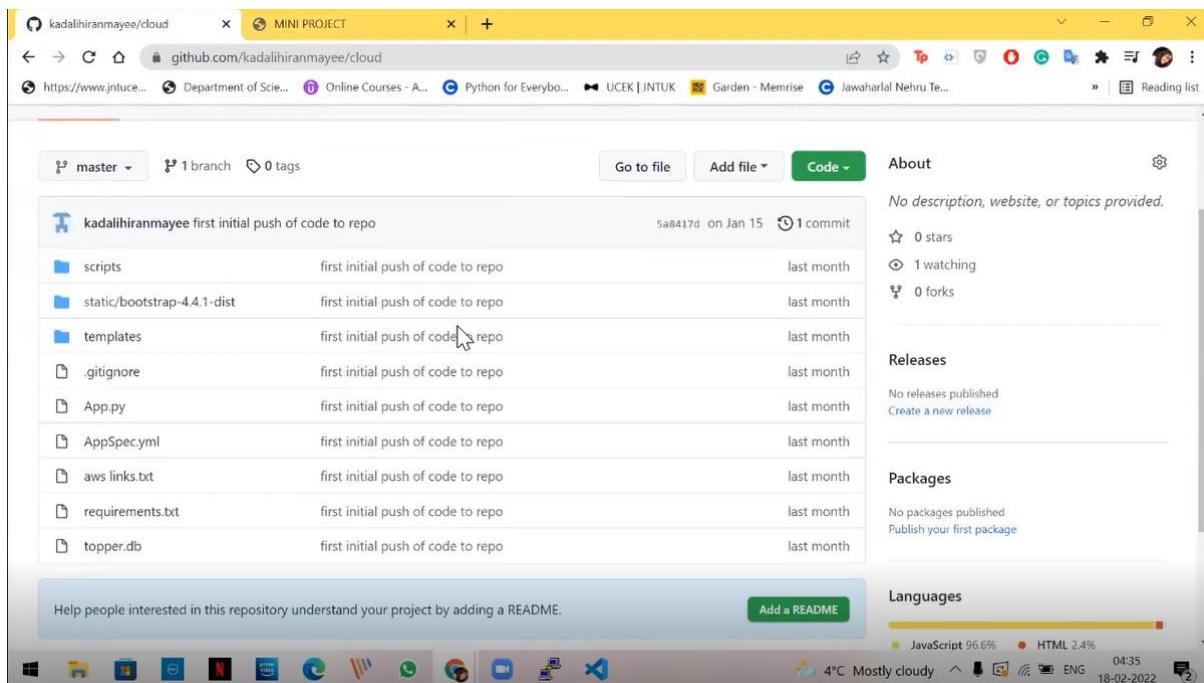
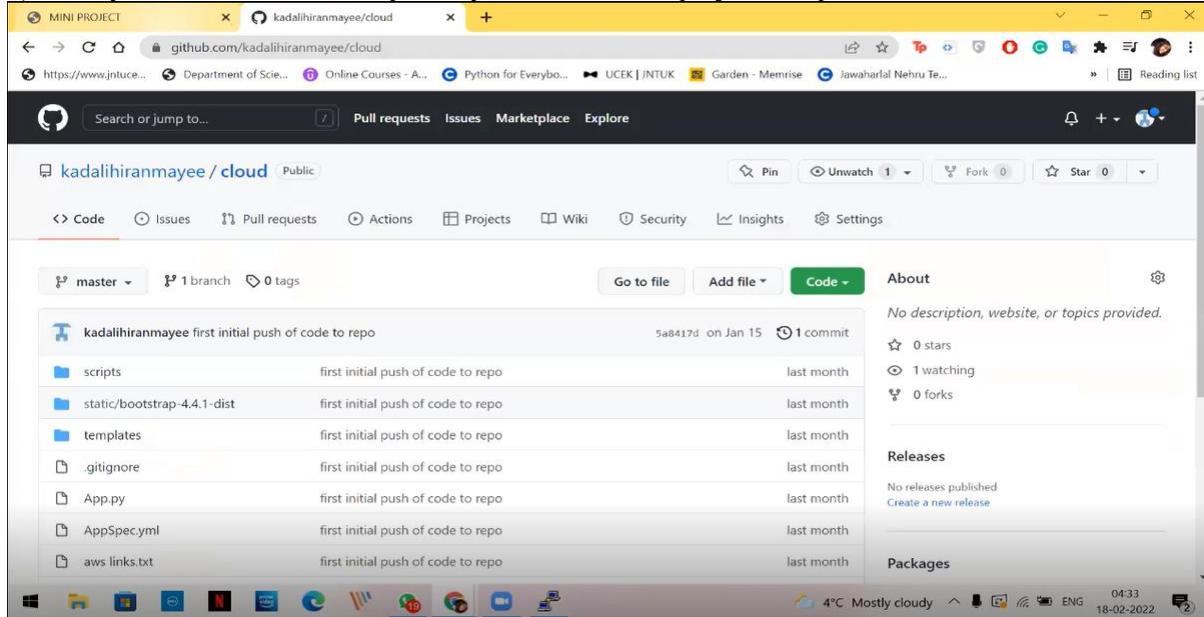
We have chosen a Flask website application for this project. Flask is web application framework written in python. We require two main packages to set the environment:

- 1) Flask
- 2) Flask – sql alchemy.

Since, using raw sql in flask web applications to perform CRUD database application can be tedious, we use SQL alchemy, which is a powerful toolkit which supports flask application with full power and flexibility.

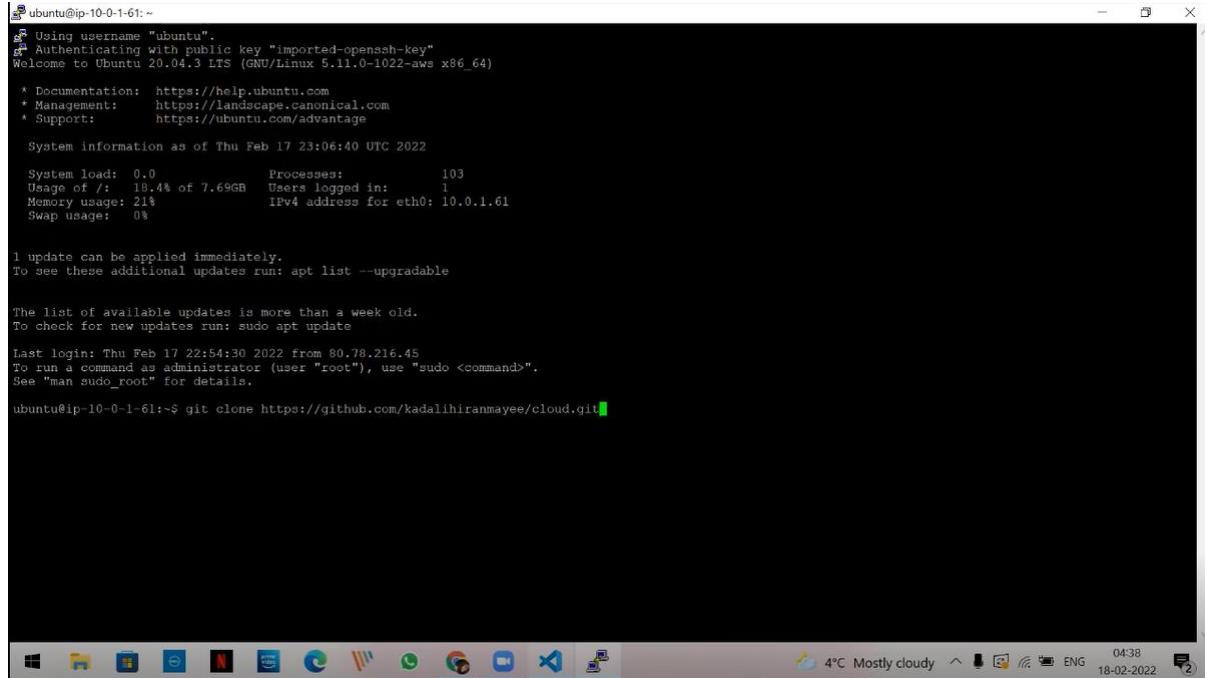
## STEPS IN DEPLOYING THE PROJECT USING UBUNTU INSTANCE:

- 1) Initially we create a GitHub repository “cloud” for our project and push all the files into it.



- 3) Now few commands are given in putty for deploying the flask application.

- **git clone https://github.com/kadalihiranmayee/cloud.git** - Cloning the git repository to access the application files.



```
ubuntu@ip-10-0-1-61:~  
Using username "ubuntu".  
Authenticating with public key "imported-openssh-key"  
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-1022-aws x86_64)  
  
* Documentation: https://help.ubuntu.com  
* Management: https://landscape.canonical.com  
* Support: https://ubuntu.com/advantage  
  
System information as of Thu Feb 17 23:06:40 UTC 2022  
  
System load: 0.0 Processes: 103  
Usage of /: 18.4% of 7.69GB Users logged in: 1  
Memory usage: 21% IPv4 address for eth0: 10.0.1.61  
Swap usage: 0%  
  
1 update can be applied immediately.  
To see these additional updates run: apt list --upgradable  
  
The list of available updates is more than a week old.  
To check for new updates run: sudo apt update  
  
Last login: Thu Feb 17 22:54:30 2022 from 80.78.216.45  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
ubuntu@ip-10-0-1-61:~$ git clone https://github.com/kadalihiranmayee/cloud.git
```

- **cd cloud** – changes the directory to cloud

```
ubuntu@ip-10-0-1-61:~/cloud
Using username "ubuntu".
Authenticating with public key "imported-openssh-key"
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-1022-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

System information as of Thu Feb 17 23:06:40 UTC 2022

System load: 0.0 Processes: 103
Usage of /: 18.4% of 7.69GB Users logged in: 1
Memory usage: 21% IPv4 address for eth0: 10.0.1.61
Swap usage: 0%

1 update can be applied immediately.
To see these additional updates run: apt list --upgradable

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Thu Feb 17 22:54:30 2022 from 80.78.216.45
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-0-1-61:~$ git clone https://github.com/kadalihiranmayee/cloud.git
Cloning into 'cloud'...
remote: Enumerating objects: 40, done.
remote: Counting objects: 100% (40/40), done.
remote: Compressing objects: 100% (27/27), done.
remote: Total 40 (delta 10), reused 40 (delta 10), pack-reused 0
Unpacking objects: 100% (40/40), 455.16 KiB | 3.58 MiB/s, done.
ubuntu@ip-10-0-1-61:~$ ls
cloud
ubuntu@ip-10-0-1-61:~$ cd cloud
ubuntu@ip-10-0-1-61:~/cloud$
```

- **ls command** – lists out all the folders in the directory.

```
ubuntu@ip-10-0-1-61:~/cloud
Using username "ubuntu".
Authenticating with public key "imported-openssh-key"
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-1022-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

System information as of Thu Feb 17 23:06:40 UTC 2022

System load: 0.0 Processes: 103
Usage of /: 18.4% of 7.69GB Users logged in: 1
Memory usage: 21% IPv4 address for eth0: 10.0.1.61
Swap usage: 0%

1 update can be applied immediately.
To see these additional updates run: apt list --upgradable

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Thu Feb 17 22:54:30 2022 from 80.78.216.45
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

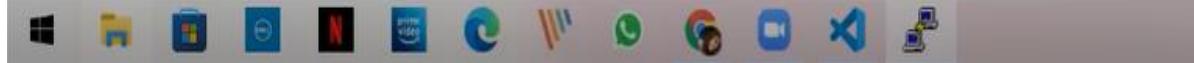
ubuntu@ip-10-0-1-61:~$ git clone https://github.com/kadalihiranmayee/cloud.git
Cloning into 'cloud'...
remote: Enumerating objects: 40, done.
remote: Counting objects: 100% (40/40), done.
remote: Compressing objects: 100% (27/27), done.
remote: Total 40 (delta 10), reused 40 (delta 10), pack-reused 0
Unpacking objects: 100% (40/40), 455.16 KiB | 3.58 MiB/s, done.
ubuntu@ip-10-0-1-61:~$ ls
cloud
ubuntu@ip-10-0-1-61:~$ cd cloud
ubuntu@ip-10-0-1-61:~/cloud$ ls
App.py AppSpec.yml 'aws links.txt' requirements.txt scripts static templates topper.db
ubuntu@ip-10-0-1-61:~/cloud$
```

- **sudo apt update** – update the ubuntu server to restore the new packages.

```

cloud
ubuntu@ip-10-0-1-61:~$ cd cloud
ubuntu@ip-10-0-1-61:~/cloud$ ls
App.py  AppSpec.yml  'aws links.txt'  requirements.txt  scripts  static  templates  topper.db
ubuntu@ip-10-0-1-61:~/cloud$ sudo apt update
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:4 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:5 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 Packages [8628 kB]
0% [5 Packages store 0 B] [Waiting for headers]

```



- **sudo apt install python3-flask** – used to install flask in the instance.

```

Building dependency tree
Reading state information... Done
69 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-10-0-1-61:~/cloud$ sudo apt install python3-flask
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  javascript-common libjs-jquery python3-itsdangerous python3-pyinotify python3-werkzeug
Suggested packages:
  apache2 | lighttpd | httpd python-flask-doc python-pyinotify-doc ipython3 python-werkzeug-doc python3-lxml python3-termcolor python3-watchdog
The following NEW packages will be installed:
  javascript-common libjs-jquery python3-flask python3-itsdangerous python3-pyinotify python3-werkzeug
0 upgraded, 6 newly installed, 0 to remove and 69 not upgraded.
Need to get 637 kB of archives.
After this operation, 2296 kB of additional disk space will be used.
Do you want to continue? [Y/n] y

```



- **sudo apt install python3-pip** - Used to install the python3 in the instance.

```

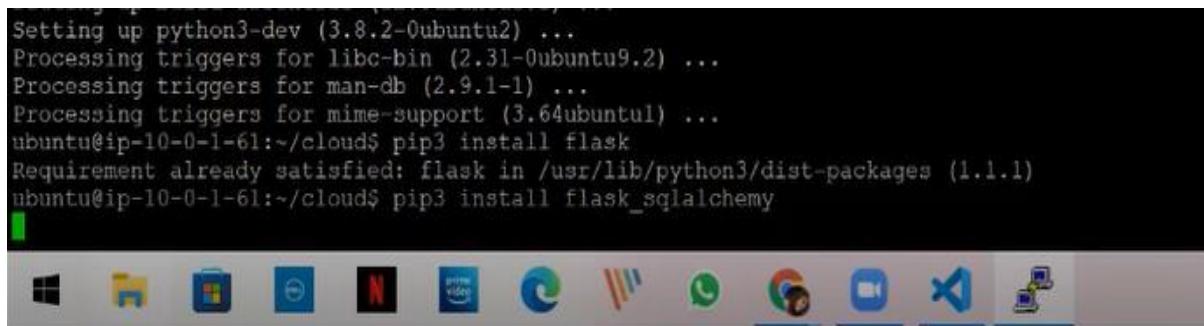
ubuntu@ip-10-0-1-61:~/cloud$ sudo apt install python3-pip
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential cpp cpp-9 dpkg-dev fakeroot g++ g++-9 gcc gcc-9-base libalgorithm-diff-perl
  libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan5 libatomic1 libbinutils libc-dev-bin libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0 libctf0
  libdpkg-perl libexpat1-dev libfakeroot libfile-fcntllock-perl libgcc-9-dev libgomp1 libis122 libitm liblsan0 libmpc3 libpython3-dev libpython3.8
  libpython3.8-dev libpython3.8-minimal libpython3.8-stdlib libquadmath0 libstdc++-9-dev libtsan0 libubsan1 linux-libc-dev make manpages-dev python-pip-whl
  python3-dev python3-wheel python3.8 python3.8-dev python3.8-minimal zlib1g-dev
Suggested packages:
  binutils-doc cpp-doc gcc-9-locales debian-keyring g++-multilib g++-9-multilib gcc-9-doc gcc-multilib autoconf automake libtool flex bison gdb gcc-doc
  gcc-9-multilib glibc-doc bzr libstdc++-9-doc make-doc python3.8-venv python3.8-doc binfmt-support
The following NEW packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential cpp cpp-9 dpkg-dev fakeroot g++ g++-9 gcc gcc-9-base libalgorithm-diff-perl
  libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan5 libatomic1 libbinutils libc-dev-bin libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0 libctf0
  libdpkg-perl libexpat1-dev libfakeroot libfile-fcntllock-perl libgcc-9-dev libgomp1 libis122 libitm liblsan0 libmpc3 libpython3-dev libpython3.8-dev
  libquadmath0 libstdc++-9-dev libtsan0 libubsan1 linux-libc-dev make manpages-dev python-pip-whl python3-dev python3-pip python3-wheel python3.8-dev
  zlib1g-dev
The following packages will be upgraded:
  libpython3.8 libpython3.8-minimal libpython3.8-stdlib python3.8 python3.8-minimal
5 upgraded, 50 newly installed, 0 to remove and 64 not upgraded.
Need to get 56.1 MB of archives.
After this operation, 214 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 python3.8 amd64 3.8.10-0ubuntu1-20.04.2 [387 kB]
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 libpython3.8 amd64 3.8.10-0ubuntu1-20.04.2 [1625 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 libpython3.8-stdlib amd64 3.8.10-0ubuntu1-20.04.2 [1675 kB]
4t [3 libpython3.8-stdlib 0 B/1675 kB 0%]

```



- **pip install flask:** This command installs flask in basic python installation.

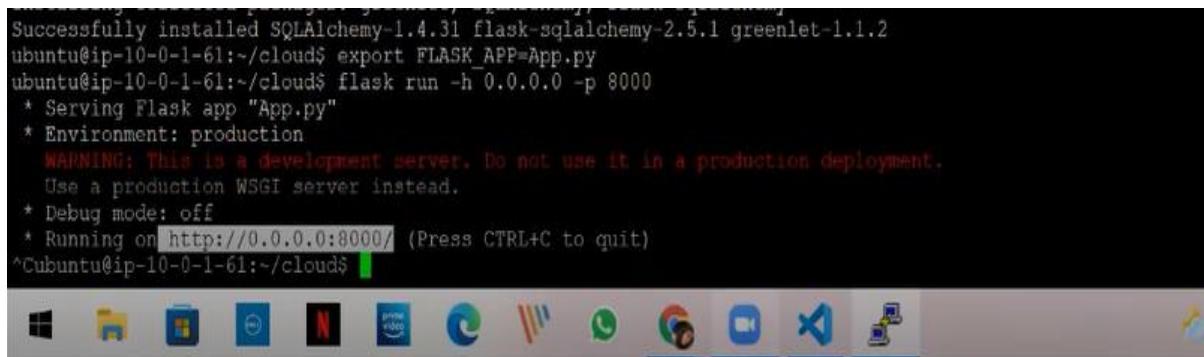
**Pip install flask sqlalchemy:** this command installs flask sqlalchemy extension in basic python installation



```
Setting up python3-dev (3.8.2-0ubuntu2) ...
Processing triggers for libc-bin (2.31-0ubuntu9.2) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for mime-support (3.64ubuntu1) ...
ubuntu@ip-10-0-1-61:~/cloud$ pip3 install flask
Requirement already satisfied: flask in /usr/lib/python3/dist-packages (1.1.1)
ubuntu@ip-10-0-1-61:~/cloud$ pip3 install flask_sqlalchemy
```

- 4) Now to run the web application on EC2 instance or virtual machine. We use the following commands.

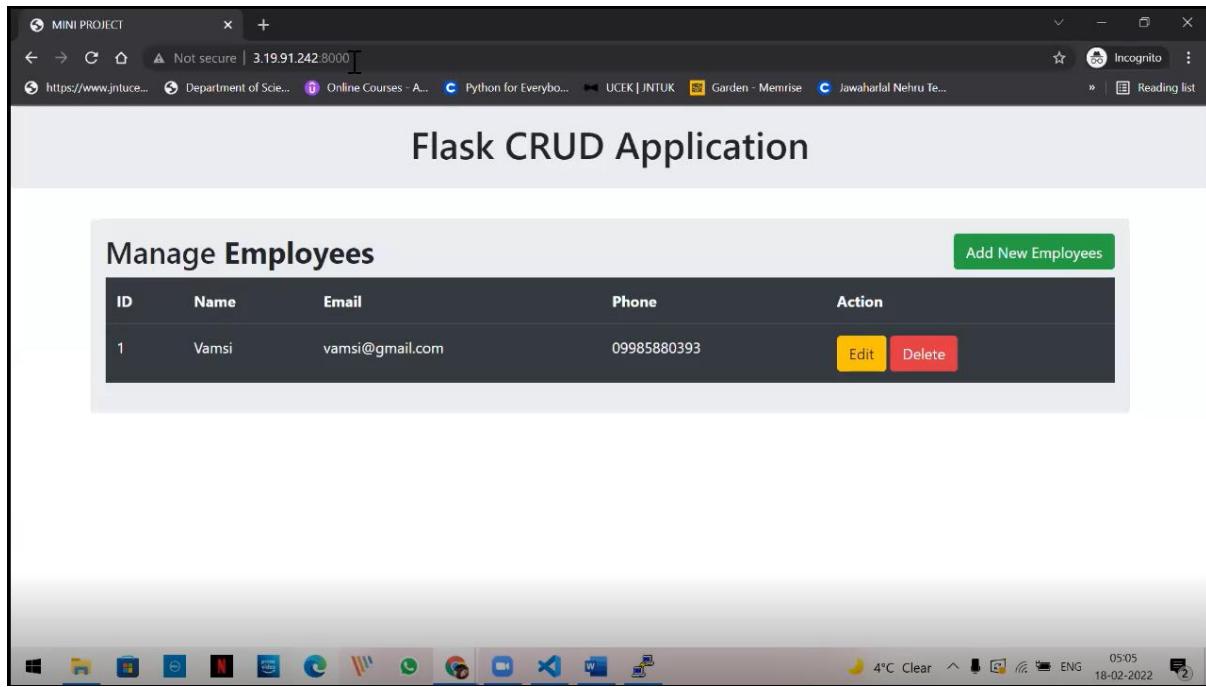
```
export FLASK_APP=APP.py
flask run -h 0.0.0.0 -p 8000
```



```
Successfully installed SQLAlchemy-1.4.31 flask-sqlalchemy-2.5.1 greenlet-1.1.2
ubuntu@ip-10-0-1-61:~/cloud$ export FLASK_APP=App.py
ubuntu@ip-10-0-1-61:~/cloud$ flask run -h 0.0.0.0 -p 8000
 * Serving Flask app "App.py"
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://0.0.0.0:8000/ (Press CTRL+C to quit)
^Cubuntu@ip-10-0-1-61:~/cloud$
```

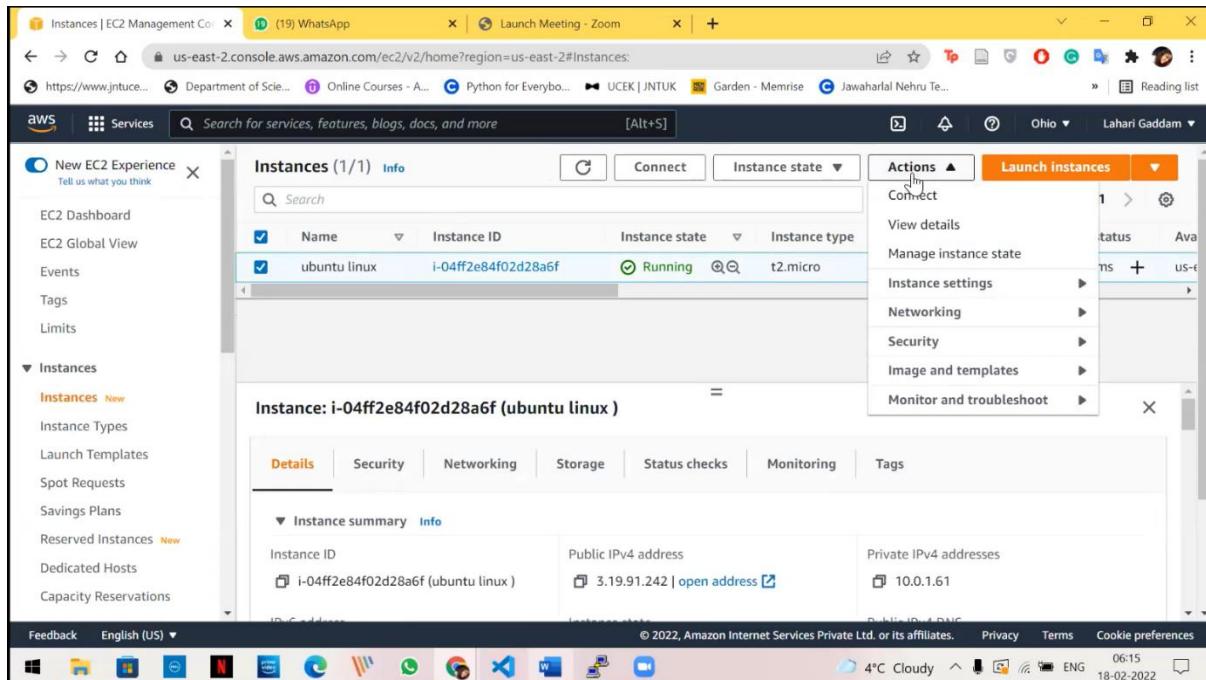
- 5) After successfully executing the code, we get a public Ip address of EC2 instance and type it on the web browser along with the port number 8000.

Our web application is successfully deployed.



## Creation of Amazon Machine Image:

An Amazon Machine Image, basically an AMI is used to give the data that is required to launch an instance. While launching an instance we need to specify the AMI for that instance. When we need to launch a greater number of instances with the same configuration then multiple instances can be launched by using a single AMI.



The screenshot shows the AWS EC2 Management Console interface. On the left, there's a sidebar with navigation links like EC2 Dashboard, EC2 Global View, Events, Tags, Limits, Instances, Instances (New), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, and Capacity Reservations. The main area displays a table titled 'Instances (1/1) Info' with one entry: Name (ubuntu linux), Instance ID (i-04ff2e84f02d28a6f), Instance state (Running), and Instance type (t2.micro). Below the table, a modal window for 'Instance: i-04ff2e84f02d28a6f (ubuntu linux)' is open. It has tabs for Details, Security, Networking, Storage, Status checks, Monitoring, and Tags. Under the Details tab, the 'Instance summary' section shows the Instance ID (i-04ff2e84f02d28a6f), Public IPv4 address (3.19.91.242), and Private IPv4 addresses (10.0.1.61). On the right side of the screen, there's a vertical Actions menu with options like Connect, View details, Manage instance state, Instance settings, Networking, Security, Image and templates, and Monitor and troubleshoot. A 'Launch instances' button is also visible.

## Creation of Launch Template:

By using launch template, you can launch an instance that shows the information of the configuration. Launch templates can be used to store the parameters to launch an instance without specifying every time when an instance is launched. Every launch template can have a greater number of launch template versions. Parameters can be different for each launch version and any of the launch template version can be set as default version.

This screenshot is identical to the one above, showing the AWS EC2 Management Console with a single running instance named 'ubuntu linux'. The interface includes the sidebar with various navigation links, the 'Instances (1/1) Info' table, the detailed instance summary modal, and the vertical Actions menu on the right.

The screenshot shows the AWS EC2 Management Console. On the left, a sidebar lists various services like EC2 Dashboard, EC2 Global View, Events, Tags, Limits, and Instances. Under Instances, 'Instances' is selected. The main pane displays a table of instances with one row selected: 'ubuntu linux' (Instance ID: i-04ff2e84f02d28a6f, Instance state: Running, Instance type: t2.micro). A context menu is open over this instance, showing options like 'Create image', 'Create template from instance' (which is highlighted with a cursor), and 'Launch more like this'. Below the table, tabs for Details, Security, Networking, Storage, Status checks, Monitoring, and Tags are visible. The 'Details' tab is active, showing the instance summary with fields for Instance ID (i-04ff2e84f02d28a6f), Public IPv4 address (3.19.91.242), and Private IPv4 addresses (10.0.1.61).

The screenshot shows the 'Create template from instance' wizard. The first step, 'Launch template name and description', is completed. It shows the source instance 'i-04ff2e84f02d28a6f' and the launch template name 'project\_20' (highlighted in a blue box). The description field contains 'A prod webserver for MyApp'. The second step, 'Launch template contents', is shown below. On the right, a summary panel displays configuration details: Software Image (AMI) - Canonical, Ubuntu, 20.04 LTS; Virtual server type (instance type) - t2.micro; Firewall (security group) - group\_20\_securitygroup; Storage (volumes) - 1 volume(s) - 8 GB. A note indicates 'Free tier: In your first year includes'. At the bottom are 'Cancel' and 'Create launch template' buttons.

The screenshot shows a browser window with the AWS EC2 console. The URL is <https://us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#CreateTemplateFromInstance:instanceId=i-04ff2...>. The page title is "Create template from instance". A success message box says "Successfully created project\_20 (lt-0fb1a0afa35c384ac)". Below it, there's a link "Actions log". Under "Next steps", there are two sections: "Launch an instance" (with a note about On-Demand Instances) and "Create an Auto Scaling group from your template" (with a note about Amazon EC2 Auto Scaling). At the bottom, there are links for "Feedback", "English (US)", and copyright information: "© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences".

## Creation of Target Group:

Target groups are created to manage the increment and decrement variations of the load on the instances. This is also ensuring high computation by avoiding network traffic.

Upon clicking the create target group, we create a target group with the basic default configuration and naming it as “project-20”.

The screenshot shows a browser window with the AWS EC2 console. The URL is <https://us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#CreateTargetGroup>. The page title is "Target groups". On the left, there's a sidebar with "New EC2 Experience" (Tell us what you think), "EC2 Dashboard", "EC2 Global View", "Events", "Tags", "Limits", and "Instances" (with sub-options: Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations). The main area shows a table titled "Target groups" with columns: Name, ARN, Port, Protocol, and Target type. A search bar at the top says "Search or filter target groups". A message at the bottom says "No target groups to display." At the bottom, there are links for "Feedback", "English (US)", and copyright information: "© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences".

Target group name: project-20

Protocol: HTTP Port: 80

VPC: group\_20

Protocol version:  HTTP1.1

Target groups have been registered.

Step 1  
Specify group details

Step 2  
Register targets

Available instances (1/1)

Instance ID	Name	State	Security groups	Zone	Subnet ID
i-04ff2e84f02d28a6f	ubuntu-linux	running	group_20_securitygroup	us-east-2a	subnet-0b21588f27807a7a1

1 selected

Ports for the selected instances  
Ports for routing traffic to the selected instances.  
80  
1-65535 (separate multiple ports with commas)

Target groups have been successfully created under the HTTP protocol.

The screenshot shows the AWS EC2 Target Groups page. At the top, a green banner says "Successfully created target group: project-20". Below it, the "Target groups (1) Info" section displays a table with one row. The table columns are Name, ARN, Port, Protocol, and Target type. The single entry is "project-20" with ARN "arn:aws:elasticloadbalancing:us-east-2:123456789012:targetgroup/project-20/1234567890123456", Port 80, Protocol HTTP, and Target type Instance. On the left sidebar, under the "Instances" section, there is a link to "Instances New".

## Configuring the Auto Scaling Group:

Auto scaling enable us to dynamically manage multiple resources on multiple servers with ease and in minutes.

Choose the template for the creation of auto scaling.

The screenshot shows the "Create template from instance" wizard, Step 4 (optional). The left sidebar lists steps: "Configure advanced options", "Step 4 (optional) Configure group size and scaling policies", "Step 5 (optional) Add notifications", "Step 6 (optional) Add tags", and "Step 7 Review". The main area shows the "Auto Scaling group name" field with "group\_20" entered. Below it, the "Launch template" dropdown is set to "project\_20". The "Description" field is empty. The "Launch template" table shows one entry: "project\_20" with ARN "arn:aws:autoscaling:us-east-2:123456789012:launchtemplate/project\_20/1234567890123456" and "Instance type t2.micro".

Selecting our vpc for the auto scaling group.

The screenshot shows the AWS EC2 Management Console with the URL <https://us-east-2.console.aws.amazon.com/ec2autoscaling/home?region=us-east-2#/create?source=launchtemplate&launch...>. The page is titled "Create template from instance".

**Step 2: Choose instance launch options**

**Network** Info

For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones. The default VPC and default subnets are suitable for getting started quickly.

**VPC**

Choose the VPC that defines the virtual network for your Auto Scaling group.

vpc-0894f2ac91c43aeda (group\_20)  
10.0.0.0/20

**Create a VPC**

**Availability Zones and subnets**

Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.

Select Availability Zones and subnets

**Instance type requirements** Info

You can keep the same instance attributes or instance type from your launch template, or you can override them.

Override launch template

Feedback English (US)

Attaching a default load balancer to the target group of the instance “group\_20”.

The screenshot shows the AWS EC2 Management Console with the URL <https://us-east-2.console.aws.amazon.com/ec2autoscaling/home?region=us-east-2#/create?source=launchtemplate&launch...>. The page is titled "Create template from instance".

**Step 2: Choose instance launch options**

**Load balancing - optional** Info

Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.

No load balancer  
Traffic to your Auto Scaling group will not be fronted by a load balancer.

Attach to an existing load balancer  
Choose from your existing load balancers.

Attach to a new load balancer  
Quickly create a basic load balancer to attach to your Auto Scaling group.

**Attach to an existing load balancer**

Select the load balancers that you want to attach to your Auto Scaling group.

Choose from your load balancer target groups  
This option allows you to attach Application, Network, or Gateway Load Balancers.

Choose from Classic Load Balancers

**Existing load balancer target groups**

Only instance target groups that belong to the same VPC as your Auto Scaling group are available for selection.

Feedback English (US)

Configuring the desired capacity, minimum capacity, and maximum capacity for the auto scaling group.

The screenshot shows the AWS Auto Scaling console during the creation of a new launch template. The current step is Step 3 (optional) titled 'Configure advanced options'. A sub-section titled 'Group size - optional' is displayed, which allows specifying the size of the Auto Scaling group by changing the desired capacity. It also lets you set minimum and maximum capacity limits. The 'Desired capacity' field contains '1', 'Minimum capacity' is '1', and 'Maximum capacity' is '2'. Below this, another section titled 'Scaling policies - optional' is visible, which is currently empty.

The screenshot shows the AWS Auto Scaling console during the creation of a new launch template. The current step is Step 6 (optional) titled 'Add tags'. A sub-section titled 'Add tags' is displayed, which allows adding tags to help search, filter, and track the Auto Scaling group across AWS. It also provides an option to automatically add these tags to instances when they are launched. A note states that you can optionally choose to add tags to instances (and their attached EBS volumes) by specifying tags in your launch template. The 'Tags (1)' section shows one tag named 'project\_20' with an 'Enter value' field containing 'I'. There is a checkbox 'Tag new instances' which is checked. Buttons for 'Add tag' and 'Remove' are present. Navigation buttons 'Cancel', 'Previous', and 'Next' are at the bottom.

An auto scaling group has been successfully created.

The screenshot shows the AWS EC2 Management Console. In the top navigation bar, there are several tabs: 'Create template from inst...', 'EC2 Management Console', 'Target groups | EC2 Man...', 'Launch Meeting - Zoom', and 'New Tab'. Below the tabs, the URL is 'us-east-2.console.aws.amazon.com/ec2autoscaling/home?region=us-east-2#details'. The browser's address bar also shows the same URL. The page title is 'Auto Scale your Amazon EC2 Instances Ahead of Demand'. A green success message at the top right says 'group\_20 created successfully. Group metrics collection is enabled.' On the left sidebar, under the 'Instances' section, 'Instances' is selected. The main content area shows a table titled 'Auto Scaling groups (1)'. The table has columns: Name, Launch template/configuration, Instances, Status, and Desired capacity. One row is listed: 'group\_20' (project\_20 | Version Default), 0 instances, 'Updating capacity', and Desired capacity 1. At the bottom of the page, there are links for 'Feedback', 'English (US)', and copyright information: '© 2022, Amazon Internet Services Private Ltd. or its affiliates.' followed by 'Privacy', 'Terms', and 'Cookie preferences'.

The screenshot shows the AWS EC2 Management Console. The top navigation bar and URL are identical to the previous screenshot. The page title is 'Instances | EC2 Management Console'. The main content area shows a table titled 'Instances (3) Info'. The table has columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability zone. Three rows are listed: 'ubuntu linux' (i-04ff2e84f02d28a6f, Running, t2.micro, 2/2 checks passed, No alarms, us-east-2), 'ubuntu linux' (i-0115679cd57fc4d8, Running, t2.micro, Initializing, No alarms, us-east-2), and 'ubuntu linux' (i-0c87d5ce82c7139b8, Running, t2.micro, 2/2 checks passed, No alarms, us-east-2). Below the table, a modal dialog is open with the title 'Select an instance'. It contains a list of the same three instances: 'ubuntu linux' (i-04ff2e84f02d28a6f), 'ubuntu linux' (i-0115679cd57fc4d8), and 'ubuntu linux' (i-0c87d5ce82c7139b8). The cursor is visible over the third item in the list.

Waiting for the auto scaling to warm up before scaling.

Filter activity history		
Status	Description	Cause
WaitingForInstanceWarmup	Launching a new EC2 instance: i-0c3b366f20478c303	At 2022-02-18T13:08:26Z a monitor alarm TargetTracking-project autoscaling-AlarmHigh-b8733eb8-7c7a-4059-93fb-e66ffcb4a0af in state ALARM triggered policy Target Tracking Policy changing the desired capacity from 1 to 2. At 2022-02-18T13:08:35Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 1 to 2.
Successful	Launching a new EC2 instance: i-040b6e65964fd30ec	At 2022-02-18T13:04:27Z a user request created an AutoScalingGroup changing the desired capacity from 0 to 1. At 2022-02-18T13:04:29Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 1.

The auto scaling group has been warmed up and is ready to scale.

Filter activity history		
Status	Description	Cause
Successful	Launching a new EC2 instance: i-0c3b366f20478c303	At 2022-02-18T13:08:26Z a monitor alarm TargetTracking-project autoscaling-AlarmHigh-b8733eb8-7c7a-4059-93fb-e66ffcb4a0af in state ALARM triggered policy Target Tracking Policy changing the desired capacity from 1 to 2. At 2022-02-18T13:08:35Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 1 to 2.
Successful	Launching a new EC2 instance: i-040b6e65964fd30ec	At 2022-02-18T13:04:27Z a user request created an AutoScalingGroup changing the desired capacity from 0 to 1. At 2022-02-18T13:04:29Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 1.

## Validation and Results:

The CRUD application has been successfully deployed on the instance using its public IP address.

Flask CRUD Application

Manage Employees

ID	Name	Email	Phone	Action
1	Vamsi	vamsi@gmail.com	09985880393	<button>Edit</button> <button>Delete</button>
2	avan	avan@gmail.com	8801135666	<button>Edit</button> <button>Delete</button>

New Employee Inserted Successfully

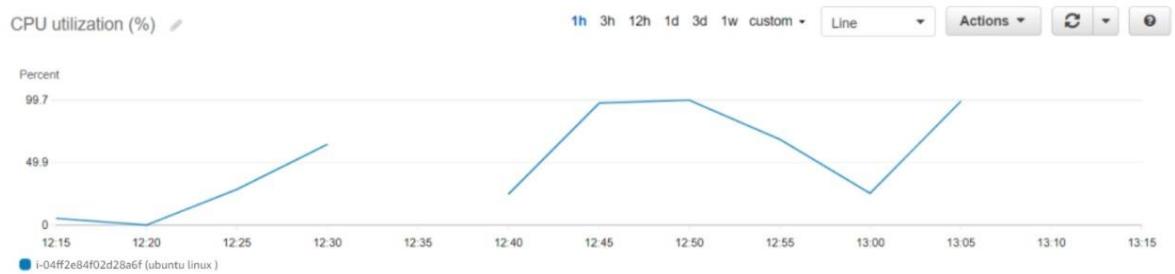
When the threshold of the load on the instance is increased, new resources are added to the instance.

Instances (3) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Ava
ubuntu linux	i-04ff2e84f02d28a6f	Running	t2.micro	2/2 checks passed	No alarms	us-e
ubuntu linux	i-0115679cd57fc4d8	Running	t2.micro	Initializing	No alarms	us-e
ubuntu linux	i-0c87d5ce82c7139b8	Running	t2.micro	2/2 checks passed	No alarms	us-e

Select an instance

Analysation of the CPU utilisation in cloud watch.



### Scalability with respect to computation:

This has been achieved by using auto scaling where resources are added dynamically whenever required. It also removes those resources that are no longer required.

### High availability of computation:

This has been achieved by configuring target groups to the instances to avoid network traffic which in turn increases the availability of the resources.