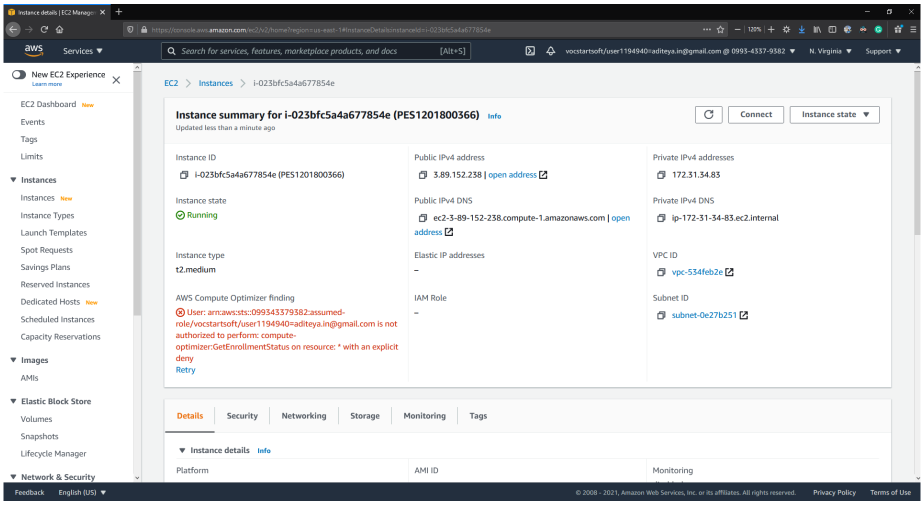
Cloud Computing Lab 2 Report

PES1201800366

Aditeya Baral

1 Web Server, Firewall Access and Monitoring Cloud Usage

1.1 Creation of AWS EC2 VMs



Instance summary for i-023bfc5a4a677854e (PES1201800366)

Updated less than a minute ago

Instance ID: i-023bfc5a4a677854e (PES1201800366)

Instance state: Running

Instance type: t2.medium

AWS Compute Optimizer finding: User: arn:aws:sts::099343379382:assumed-role/vocstartsoft/user1194940=aditeya.in@gmail.com is not authorized to perform: compute-optimizer:GetEnrollmentStatus on resource: * with an explicit deny

Retry

Public IPv4 address: 3.89.152.238 | open address

Public IPv4 DNS: ec2-3-89-152-238.compute-1.amazonaws.com | open address

Elastic IP addresses: –

IAM Role: –

Private IPv4 addresses: 172.31.34.83

Private IPv4 DNS: ip-172-31-34-83.ec2.internal

VPC ID: vpc-534feb2e

Subnet ID: subnet-0e27b251

▼ Instance details Info

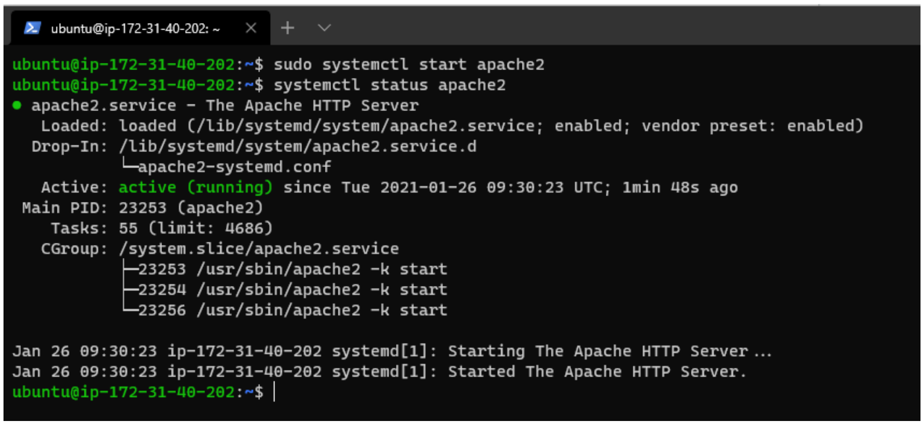
Platform

AMI ID

Monitoring

Figure 1: Creation of the EC2 Virtual Machine

1.2 Installation of Apache2 Web Server

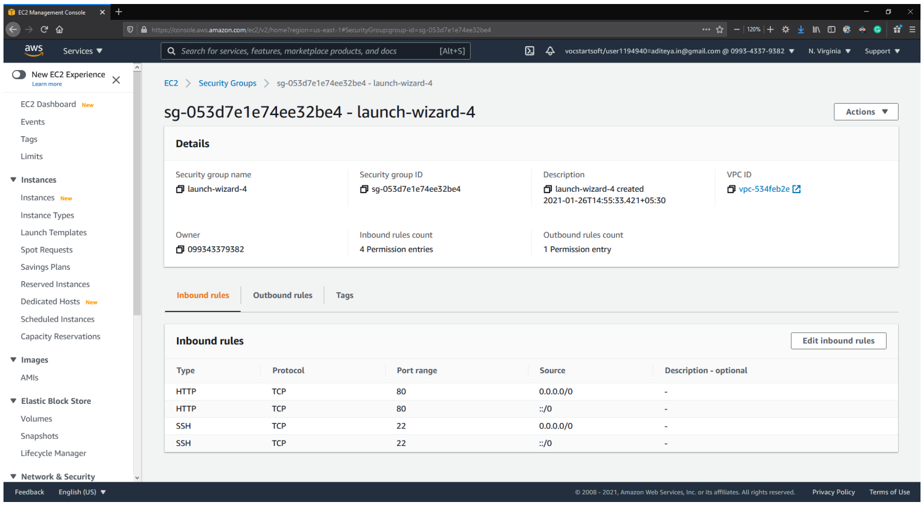


```
ubuntu@ip-172-31-40-202:~$ sudo systemctl start apache2
ubuntu@ip-172-31-40-202:~$ systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
  Drop-In: /lib/systemd/system/apache2.service.d
           └─apache2-systemd.conf
   Active: active (running) since Tue 2021-01-26 09:30:23 UTC; 1min 48s ago
 Main PID: 23253 (apache2)
    Tasks: 55 (limit: 4686)
   CGroup: /system.slice/apache2.service
           ├─23253 /usr/sbin/apache2 -k start
           ├─23254 /usr/sbin/apache2 -k start
           └─23256 /usr/sbin/apache2 -k start

Jan 26 09:30:23 ip-172-31-40-202 systemd[1]: Starting The Apache HTTP Server...
Jan 26 09:30:23 ip-172-31-40-202 systemd[1]: Started The Apache HTTP Server.
ubuntu@ip-172-31-40-202:~$ |
```

Figure 2: Installing and starting Apache2 Web Server

1.3 Setup Firewall Rules



sg-053d7e1e74ee32be4 - launch-wizard-4

Details

Security group name: launch-wizard-4

Security group ID: sg-053d7e1e74ee32be4

Description: launch-wizard-4 created 2021-01-26T14:55:33.421+05:30

VPC ID: vpc-534feb2e

Owner: 099343379382

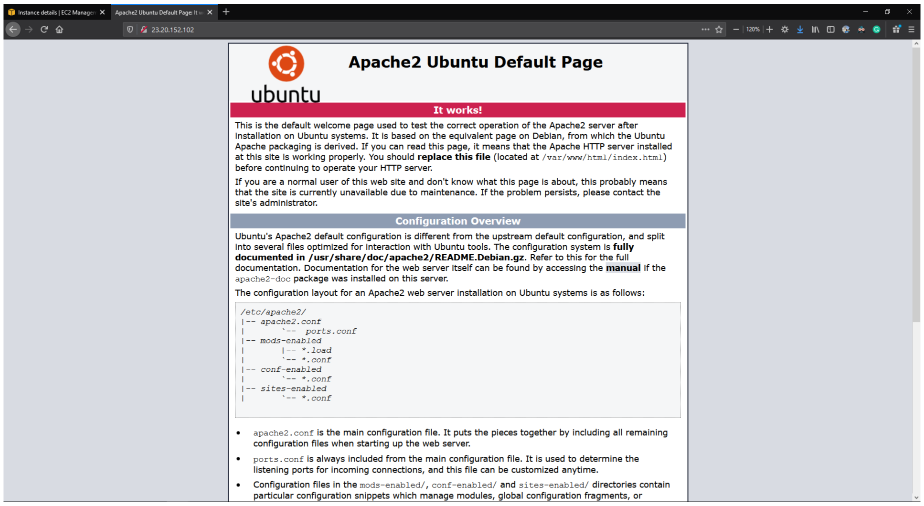
Inbound rules count: 4 Permission entries

Outbound rules count: 1 Permission entry

Inbound rules

Type	Protocol	Port range	Source	Description - optional
HTTP	TCP	80	0.0.0.0/0	-
HTTP	TCP	80	::/0	-
SSH	TCP	22	0.0.0.0/0	-
SSH	TCP	22	::/0	-

Figure 3: Setting Firewall rules to allow HTTP access


Instance details | EC2 Managem
Apache2 Ubuntu Default Page: It w
23.20.152.102
120%
ubuntu

Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

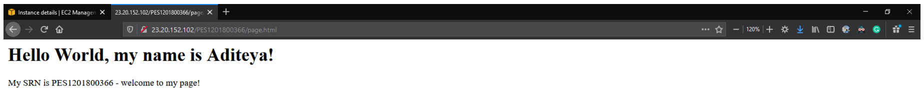
The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or

Figure 4: Accessing default Apache2 page on local machine

1.4 Hosting a Web Page

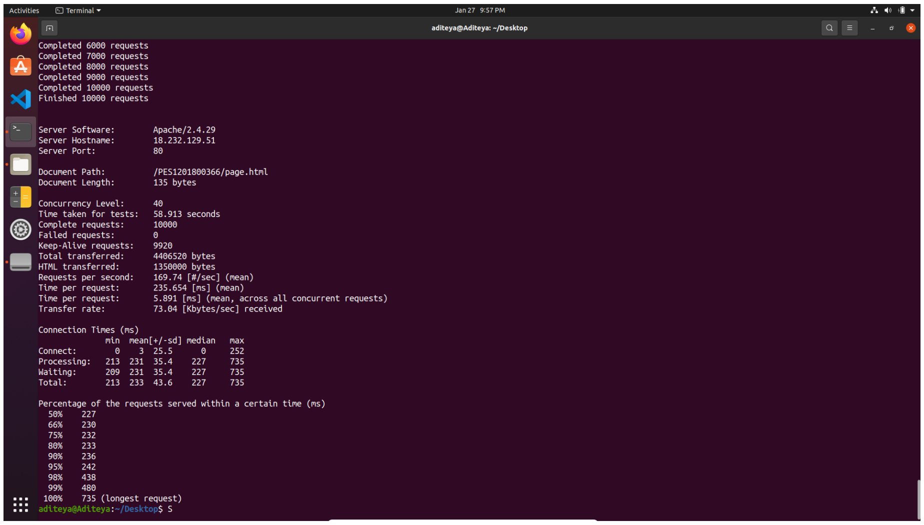

Instance details | EC2 Managem
23.20.152.102/PES1201800366/page
23.20.152.102/PES1201800366/page.html
120%

Hello World, my name is Aditeya!

My SRN is PES1201800366 - welcome to my page!

Figure 5: Hosting custom web page using Apache2

1.5 ApacheBench and Cloudwatch



```
Completed 6000 requests
Completed 7000 requests
Completed 8000 requests
Completed 9000 requests
Completed 10000 requests
Finished 10000 requests


Server Software:        Apache/2.4.29
Server Hostname:        18.232.129.51
Server Port:            80

Document Path:          /PES1201800366/page.html
Document Length:        135 bytes

Concurrency Level:      40
Time taken for tests:   58.913 seconds
Complete requests:      10000
Failed requests:        0
Keep-Alive requests:    9920
Total transferred:      4406520 bytes
HTML transferred:       1350000 bytes
Requests per second:    169.74 [#/sec] (mean)
Time per request:       235.654 [ms] (mean)
Time per request:       5.891 [ms] (mean, across all concurrent requests)
Transfer rate:          73.04 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0    3  25.5      0     252
Processing:   213  231  35.4    227     735
Waiting:      209  231  35.4    227     735
Total:        213  233  43.6    227     735

Percentage of the requests served within a certain time (ms)
  50%    227
  66%    230
  75%    232
  80%    233
  90%    236
  95%    242
  98%    438
  99%    480
 100%    735 (longest request)
aditeya@Aditeya:~/Desktop$ S
```

Figure 6: Using ApacheBench to create loads on web server

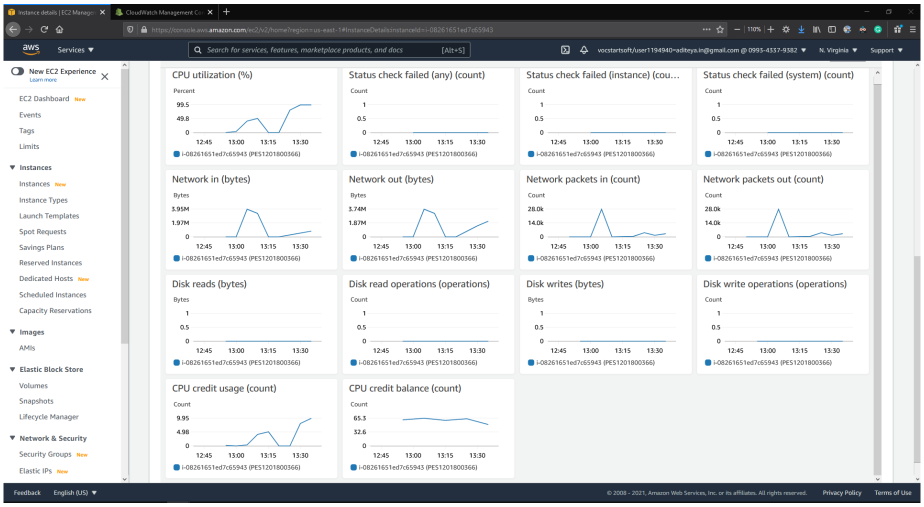
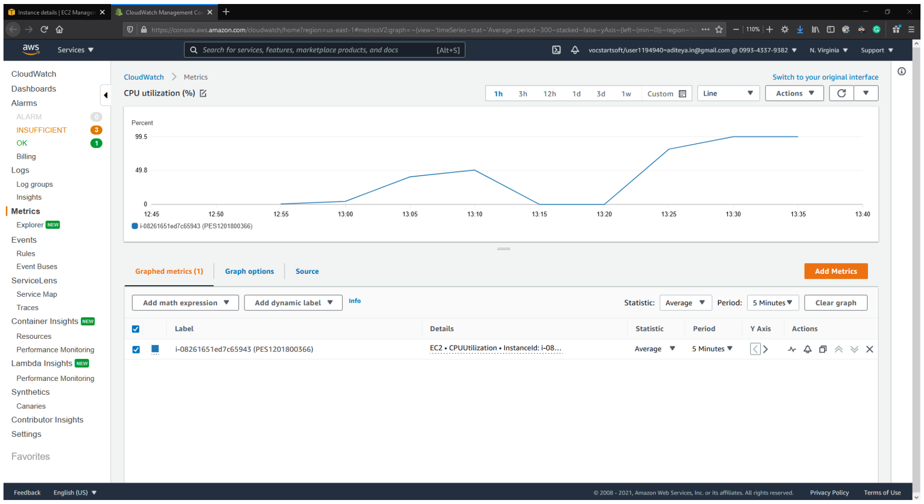

Instance details | EC2 Manage
CloudWatch Management Con
https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#InstanceDetails:instanceId=i-08261651ed7c65943
110%
aws
Services
Search for services, features, marketplace products, and docs [Alt+S]
vocstartsoft/user1194940=aditeya.in@gmail.com @ 0993-4337-9382
N. Virginia
Support
New EC2 Experience
Learn more
EC2 Dashboard New
Events
Tags
Limits
Instances
Instances New
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts New
Scheduled Instances
Capacity Reservations
Images
AMIs
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Network & Security
Security Groups New
Elastic IPs New
CPU utilization (%)
Percent
99.5
49.8
0
Status check failed (any) (count)
Status check failed (instance) (cou…
Status check failed (system) (count)
Count
1
0.5
0
Network in (bytes)
Bytes
3.95M
1.97M
Network out (bytes)
3.74M
1.87M
Network packets in (count)
Network packets out (count)
28.0k
14.0k
Disk reads (bytes)
Disk read operations (operations)
Disk writes (bytes)
Disk write operations (operations)
CPU credit usage (count)
9.95
4.98
CPU credit balance (count)
65.3
32.6
12:45
13:00
13:15
13:30
i-08261651ed7c65943 (PES1201800366)
Feedback
English (US)
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

Figure 7: Plots of instance metrics after creating web server load



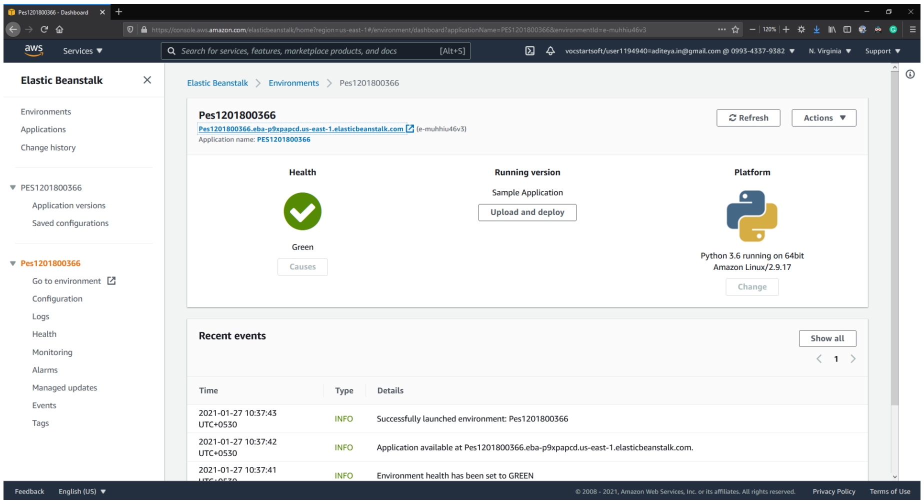
Statistic: Average Period: 5 Minutes

☑		Label	Details	Statistic	Period	Y Axis	Actions
☑	🟦	i-08261651ed7c65943 (PES1201800366)	EC2 • CPUUtilization • InstanceId: i-08...	Average	5 Minutes		

Figure 8: Observing CPU utilisation after adding CPU stress load

2 Deploying Flask App on AWS Elastic Beanstalk

2.1 Launching AWS Beanstalk Environment



Pes1201800366

Pes1201800366.eba-p9xpapcd.us-east-1.elasticbeanstalk.com (e-muhhiu46v3)

Application name: PES1201800366

Health: Green

Running version: Sample Application

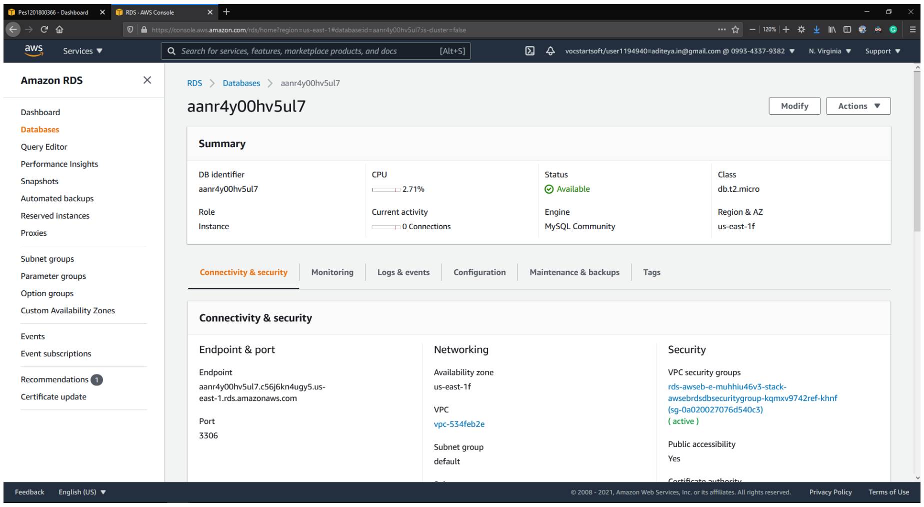
Platform: Python 3.6 running on 64bit Amazon Linux/2.9.17

Recent events

Time	Type	Details
2021-01-27 10:37:43 UTC+0530	INFO	Successfully launched environment: Pes1201800366
2021-01-27 10:37:42 UTC+0530	INFO	Application available at Pes1201800366.eba-p9xpapcd.us-east-1.elasticbeanstalk.com.
2021-01-27 10:37:41 UTC+0530	INFO	Environment health has been set to GREEN

Figure 9: Creation of AWS Beanstalk Environment

2.2 Adding AWS RDS Database



https://console.aws.amazon.com/rds/home?region=us-east-1#database:id=aanr4y00hv5ul7;is-cluster=false

vocstartsoft/user1194940=aditeya.in@gmail.com @ 0993-4337-9382

RDS > Databases > aanr4y00hv5ul7

aanr4y00hv5ul7

Summary

DB identifier	CPU	Status	Class
aanr4y00hv5ul7	2.71%	Available	db.t2.micro
Role	**Current activity**	**Engine**	**Region & AZ**
Instance	0 Connections	MySQL Community	us-east-1f

Connectivity & security

Endpoint & port

Endpoint
aanr4y00hv5ul7.c56j6kn4ugy5.us-east-1.rds.amazonaws.com

Port
3306

Networking

Availability zone
us-east-1f

VPC
vpc-534feb2e

Subnet group
default

Security

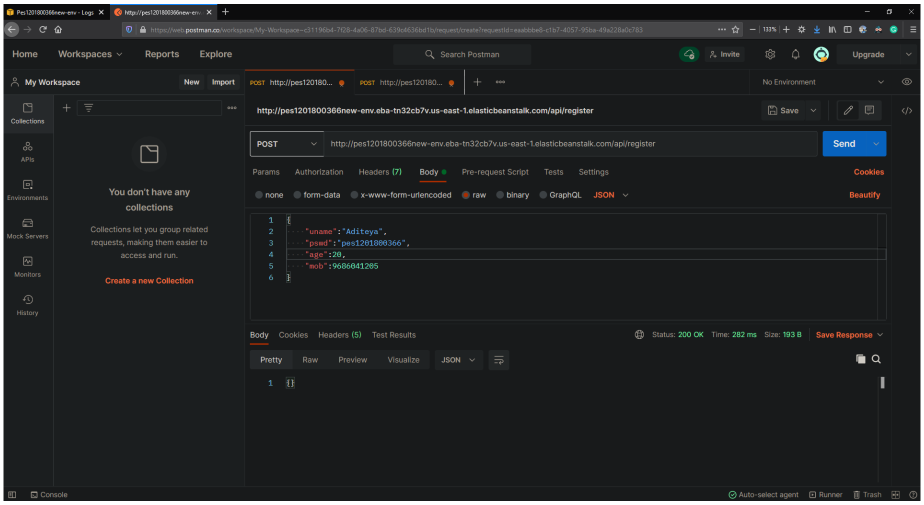
VPC security groups
rds-awseb-e-muhhiu46v3-stack-awsebrdsdbsecuritygroup-kqmxv9742ref-khnf (sg-0a020027076d540c3)
(active)

Public accessibility
Yes

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Figure 10: Adding RDS Database to environment

2.3 Deploy Flask App and usage of REST APIs



https://web.postman.co/workspace/My-Workspace~c31196b4-7f28-4a06-87bd-639c4636bd1b/request/create?requestId=eaabbbe8-c1b7-4057-95ba-49a228a0c783

You don't have any collections

Collections let you group related requests, making them easier to access and run.

Create a new Collection

http://pes1201800366new-env.eba-tn32cb7v.us-east-1.elasticbeanstalk.com/api/register

POST http://pes1201800366new-env.eba-tn32cb7v.us-east-1.elasticbeanstalk.com/api/register

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

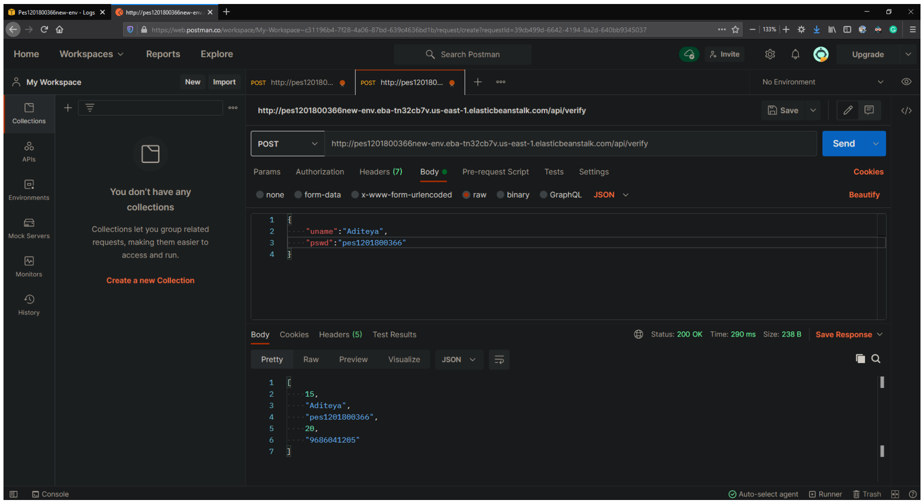
```
{
    "uname":"Aditeya",
    "pswd":"pes1201800366",
    "age":20,
    "mob":9686041205
}
```

Body

Status: 200 OK Time: 282 ms Size: 193 B

```
{}
```

Figure 11: Registering user on Postman



http://pes1201800366new-env.eba-tn32cb7v.us-east-1.elasticbeanstalk.com/api/verify

POST http://pes1201800366new-env.eba-tn32cb7v.us-east-1.elasticbeanstalk.com/api/verify

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
{
    "uname":"Aditeya",
    "pswd":"pes1201800366"
}
```

Status: 200 OK Time: 290 ms Size: 238 B

```
[
    15,
    "Aditeya",
    "pes1201800366",
    20,
    "9686041205"
]
```

Figure 12: Verifying user on Postman