

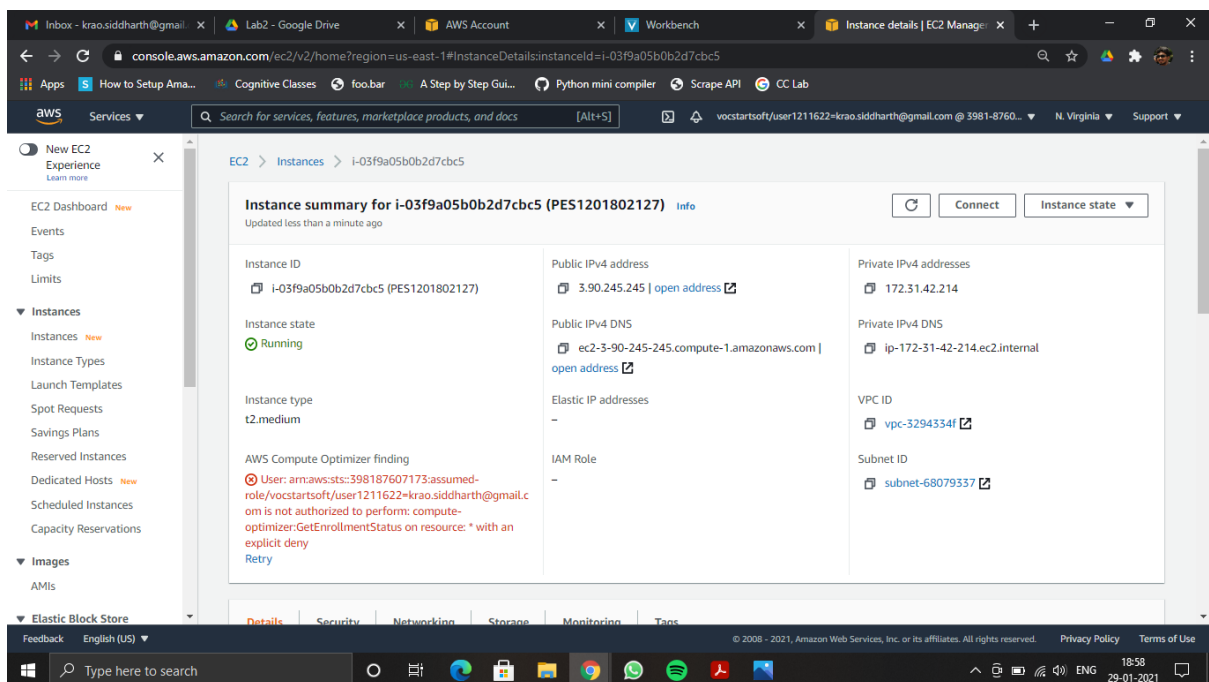
# CLOUD COMPUTING LAB: WEEK 2

PES1201802127

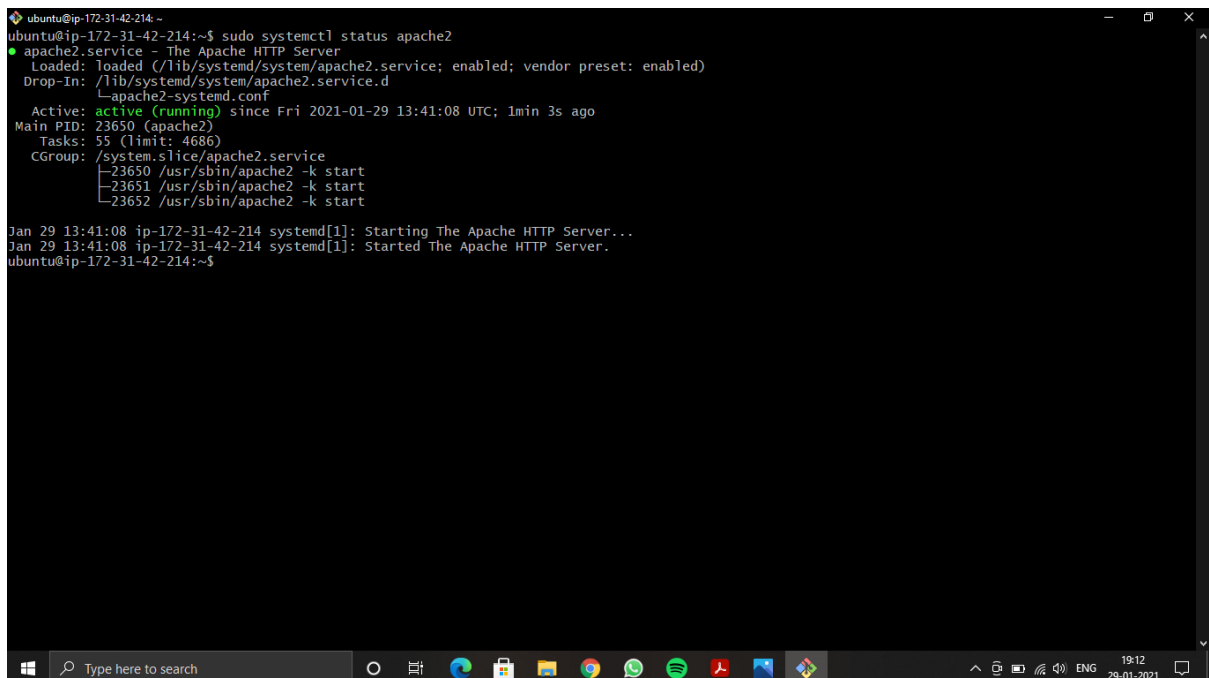
Siddharth K Rao

- 1.) Task A: The Objective of this task is to setup up a web server on the AWS EC2 Instance and host a sample web page on the instance. You are also required to load test the web page by using Apache Bench and monitor the EC2 instance using AWS CloudWatch service.

## 1a) Creating EC2 Instance:



## 2a) Installing Apache web Server on AWS EC2 Instance:



### 3a) Setting up Firewall rules and showing the list of rules updated:

The screenshot shows the AWS Management Console interface. The left sidebar contains navigation options like 'New EC2 Experience', 'EC2 Dashboard', 'Events', 'Tags', 'Limits', 'Instances', 'Images', and 'Elastic Block Store'. The main content area displays the details for a security group named 'launch-wizard-1' with ID 'sg-0021062a236c21dde'. The details include the security group name, ID, description, VPC ID, owner, inbound rules count (3), and outbound rules count (1). Below the details, there are tabs for 'Inbound rules', 'Outbound rules', and 'Tags'. The 'Inbound rules' tab is active, showing a table of inbound rules.

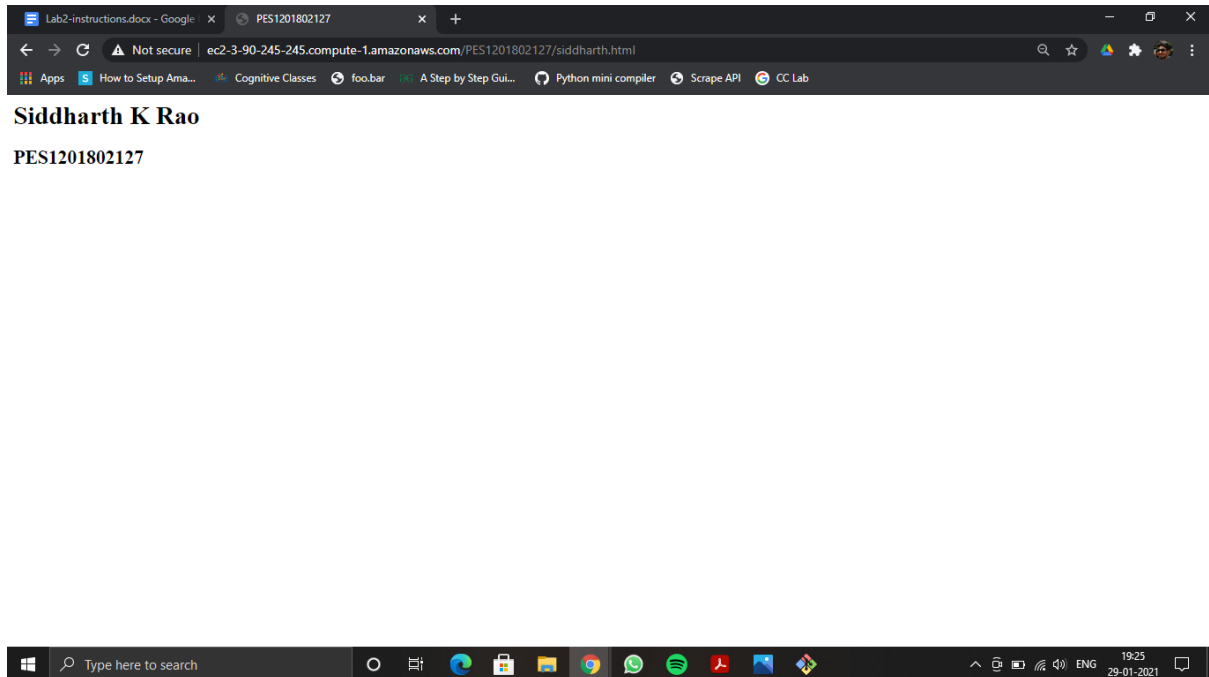
Type	Protocol	Port range	Source	Description - optional
HTTP	TCP	80	0.0.0.0/0	-
HTTP	TCP	80	:::0	-
SSH	TCP	22	0.0.0.0/0	-

### 3b) Local browser showing EC2 Instance's Apache2 Ubuntu Default Page:

The screenshot shows a local browser window displaying the 'Apache2 Ubuntu Default Page'. The page has a red header with the Ubuntu logo and the text 'It works!'. Below the header, there is a paragraph of text explaining that this is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It also mentions that if the page is not visible, it might be due to maintenance or a problem with the site's administrator. Below the text, there is a section titled 'Configuration Overview' which provides information about the Apache2 default configuration on Ubuntu, including the location of the configuration files and the main configuration file.

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

#### 4a) Hosting a sample web page:



#### 5a) Apachebench output report after execution:

```
ubuntu@ip-172-31-42-214: ~$ ab -n 10000 http://ec2-3-90-245-245.compute-1.amazonaws.com/PES1201802127/siddharth.html
Finished 10000 requests

Server Software:         Apache/2.4.29
Server Hostname:         ec2-3-90-245-245.compute-1.amazonaws.com
Server Port:             80

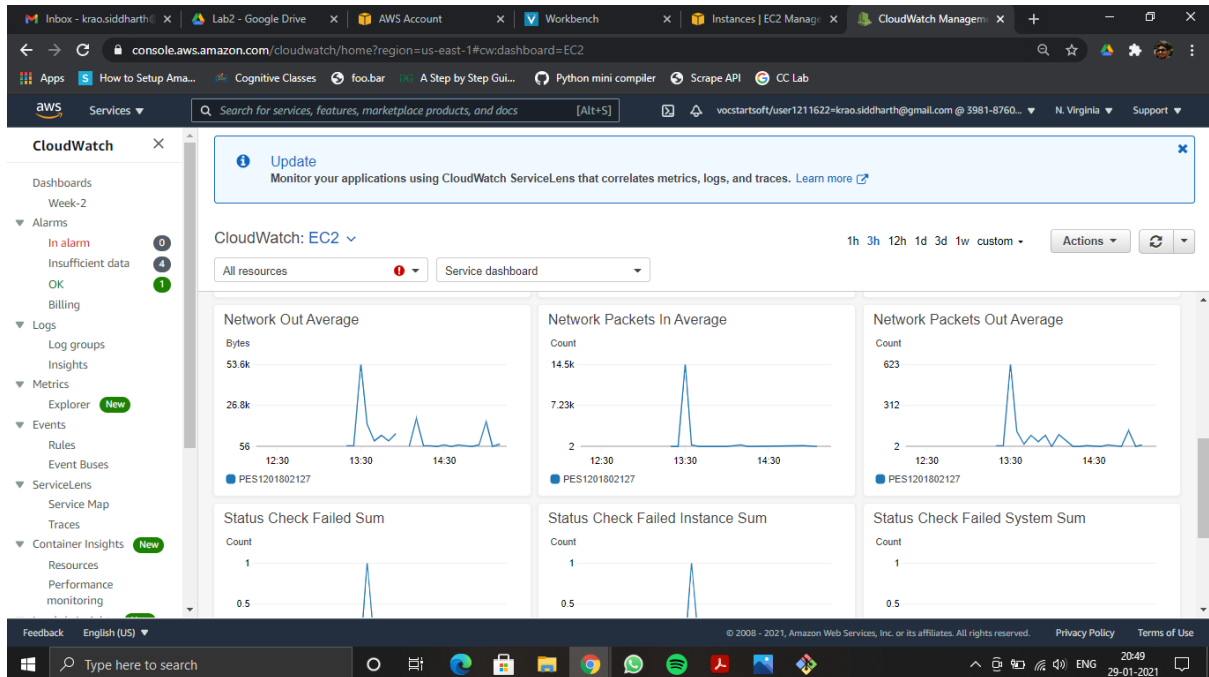
Document Path:           /PES1201802127/siddharth.html
Document Length:         132 bytes

Concurrency Level:       40
Time taken for tests:    0.721 seconds
Complete requests:       10000
Failed requests:         355
    (Connect: 0, Receive: 0, Length: 355, Exceptions: 0)
Keep-Alive requests:     9611
Total transferred:      4223386 bytes
HTML transferred:       1273140 bytes
Requests per second:    13874.71 [#/sec] (mean)
Time per request:       2.883 [ms] (mean)
Time per request:       0.072 [ms] (mean, across all concurrent requests)
Transfer rate:          5722.48 [Kbytes/sec] received

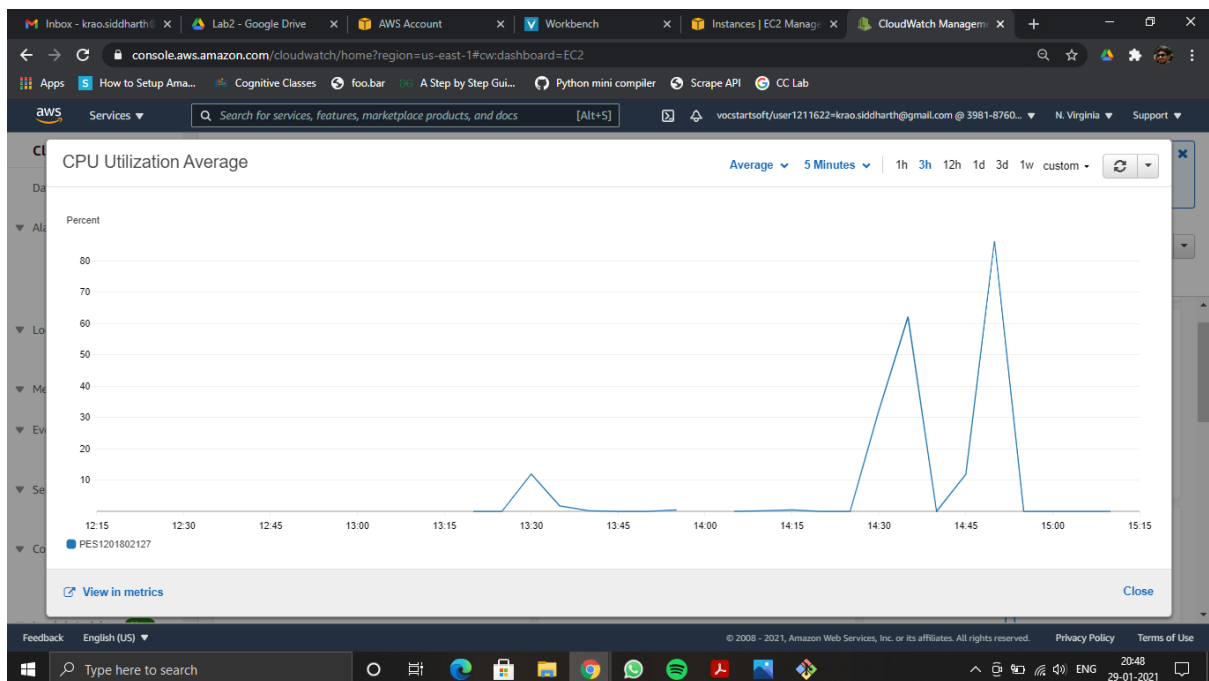
Connection Times (ms)
  min   mean[+/-sd] median   max
Connect:    0    0   0.2      0     4
Processing:  0    3   2.8      2    46
Waiting:    0    3   2.8      2    46
Total:      0    3   2.8      2    46

Percentage of the requests served within a certain time (ms)
 50%    2
 66%    3
 75%    3
 80%    4
 90%    5
 95%    7
 98%   10
 99%   13
100%   46 (longest request)
ubuntu@ip-172-31-42-214:~$ ab -k -c 40 -n 10000 http://ec2-3-90-245-245.compute-1.amazonaws.com/PES1201802127/siddharth.html
```

## 5b) All set of graphs:



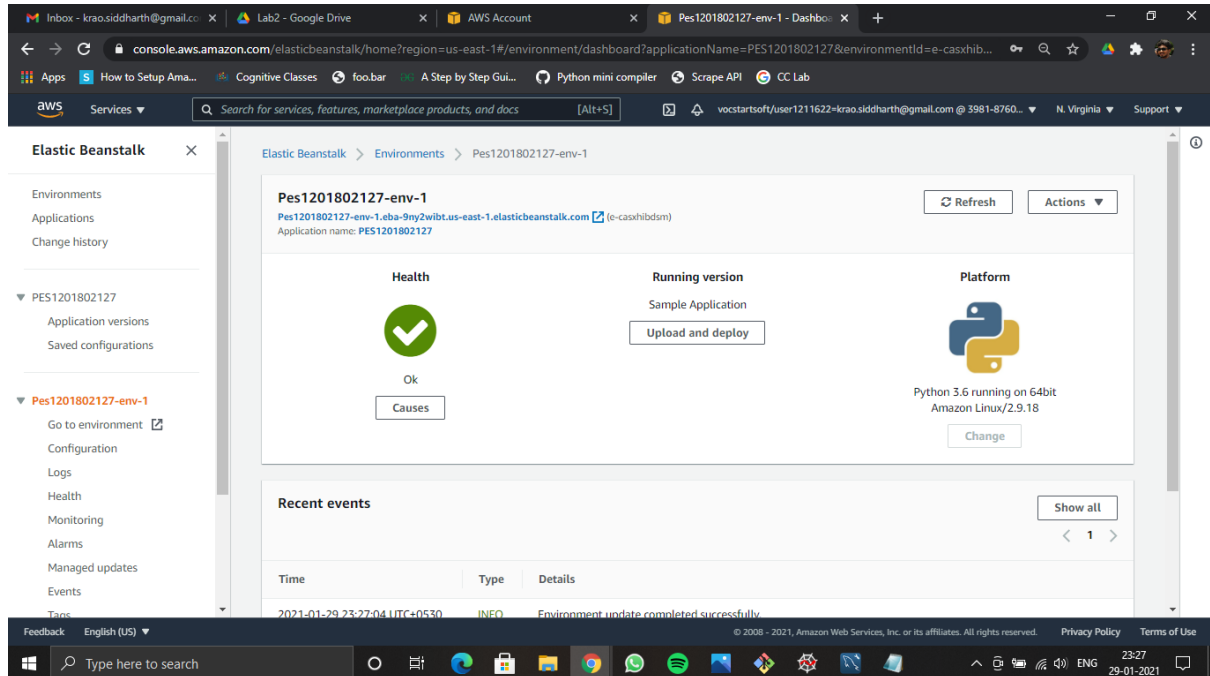
## 5c) CPU Utilization:



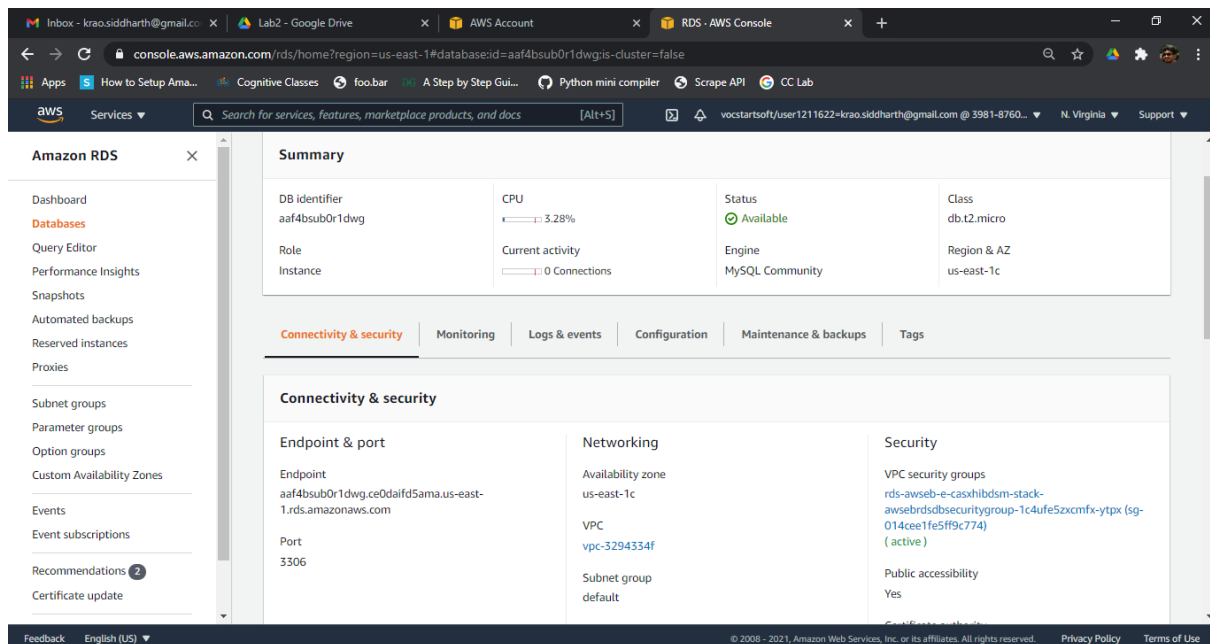
## 2.) Task 2: Deploying Flask app On AWS Elastic Beanstalk

In this task you'll be asked to create a AWS Elastic Beanstalk environment and deploy the flask application provided to you. You are required to configure a RDS along with your Beanstalk environment and create a few tables and databases.

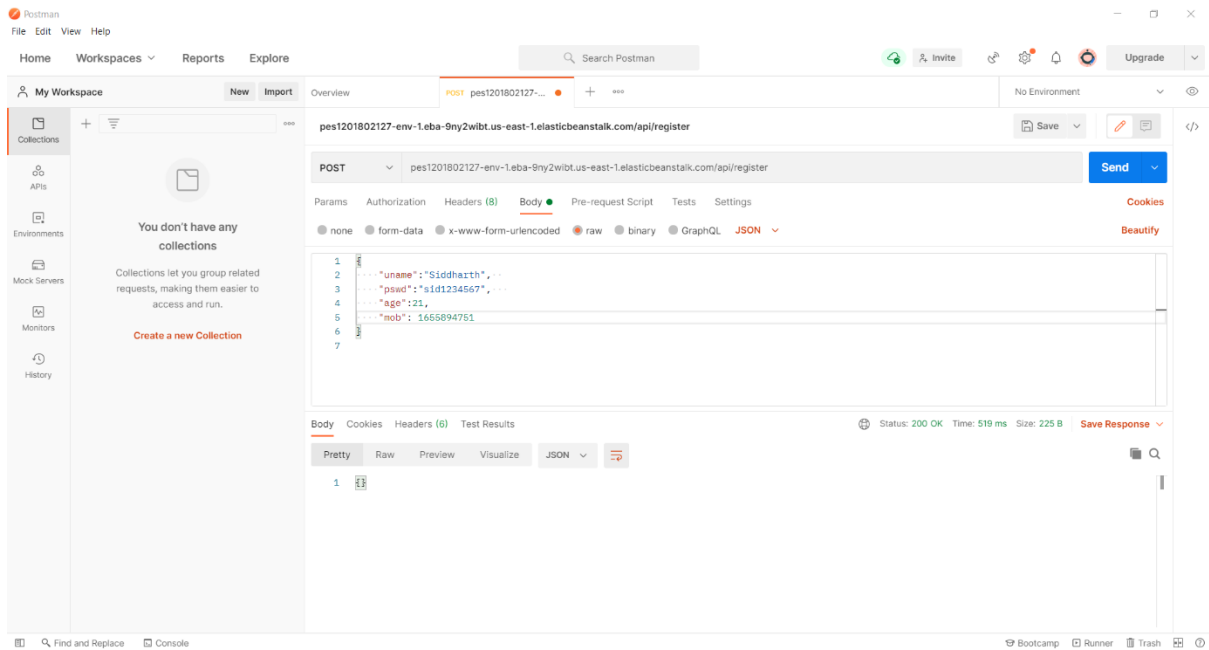
### B1) Environment Health:



### B2) DB Configuration:



## B3) Register User Request:



## B4) Verify User Request:

