INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, GUWAHATI

CS351 PROJECT

INTELLIGENT HYPERVISOR

1. AILNENI SAI SHISHIR (1701002)
2. ANIVRAT GOEL (1701008)
3. SUSHANT AGGARWAL (1701069)

ABSTRACT:

Intelligent hypervisor design: In the traditional cloud world, VMs are created when requested to the hypervisor. Design a solution where VMs are created by an intelligent system monitors the eco system, and requests the hypervisor to create VMs in order to comply with service level agreement of processing speed / turnaround etc.

- In the above problem statement we were asked to create an hypervisor, which is a very big task and cannot be done without specialized hardware. Hence, we have simulated the task of hypervisor on Amazon Web Services.

- In AWS we consider the server as our system and monitor the environment. When any parameters are not satisfied we launch instances to actually comply with the SLA.

- Each instance is considered as a separate virtual machine.

- This process can be achieved by AWS AutoScaling in which we can monitor the system and can launch instances according to it.

STEPS INVOLVED IN THE PROCESS

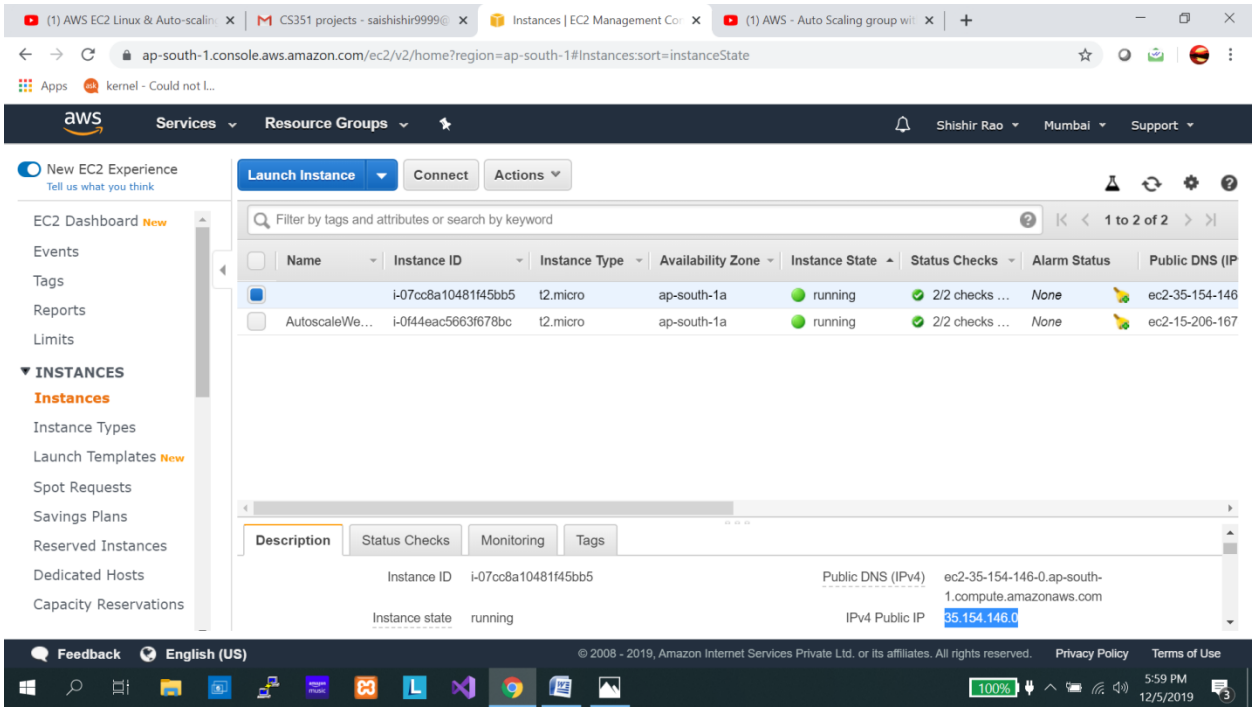
1. **CREATE A WEBSERVER IN AWS.**
2. **CREATE AN IMAGE OF THE WEBSERVER.**
3. **CREATE AN AUTOSCALING GROUP.**
4. **APPLY SCALE UP AND SCALE DOWN POLICIES.**
5. **CREATE NETWORK/SUBNETS.**
6. **CREATE ALARMS.**
7. **OPEN YOUR INSTANCE FROM TERMINAL USING SSH KEY.**
8. **EXECUTE A SCRIPT TO INCREASE THE LOAD ON CPU**
9. **CHECK THE OUTPUT, WE CAN SEE NUMBER OF RUNNING INSTANCES INCREASED.**

DISCLAIMER:-

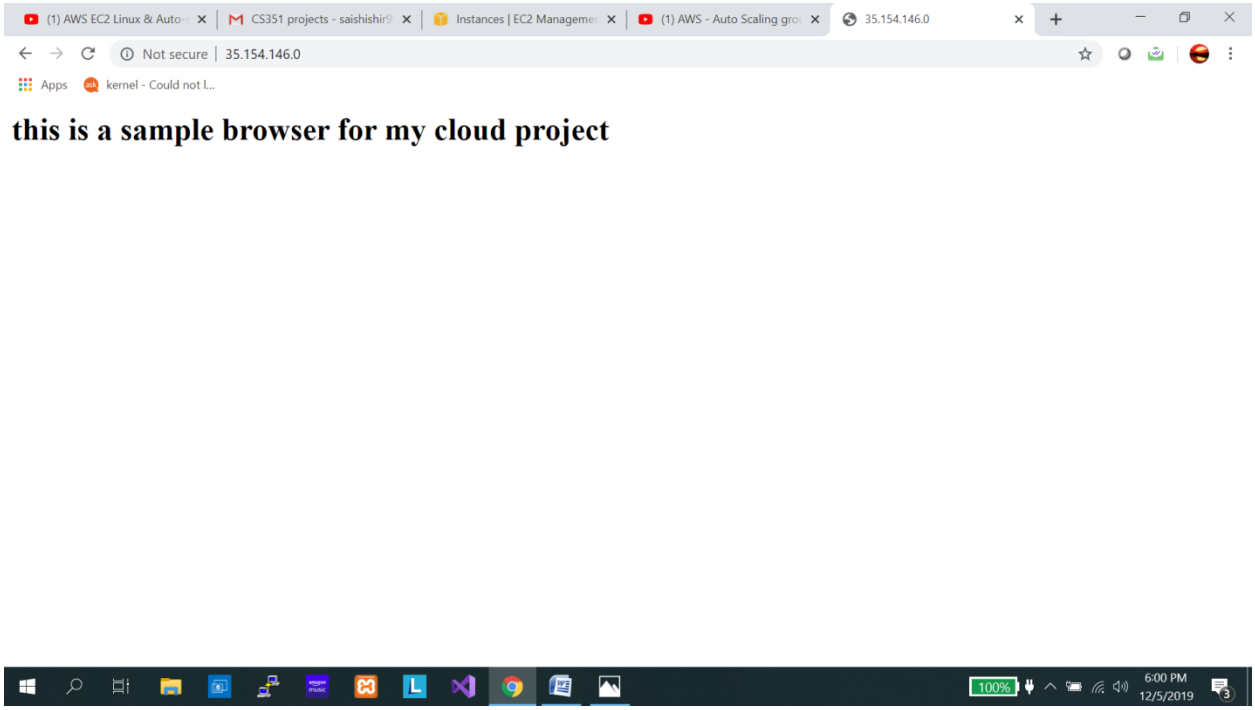
1. **ALL THE CODES GIVEN BELOW ARE USING OUR KEYS AND OUR SYSTEMS. PLEASE DO CHANGE THEM TO EXECUTE THE CODES.**
2. **PYTHON 3 IS REQUIRED TO EXECUTE THE CODES.**
3. **FOR CONVENIENCE PLZ GO VIA GRAPHICAL APPROACH WHICH WILL BE A BIT EASY.**

STEP-1:-

- **GO TO EC2 IN AWS**
- **CLICK LAUNCH INSTANCE**
- **SELECT A LINUX INSTANCE**
- **SET TO T2.MICRO**
- **IN INSTANCE DETAILS CLICK ON ADVANCE AND PASTE THE FOLLOWING SCRIPT.**
- **#!/bin/bash**
- **Yum install httpd –y**
- **Echo '<h1>this is a sample website for cloud project<h1>' > /var/www/html/index.html**
- **Chconfig httpd on**
- **Service httpd start**
- **AFTER ENTERING THE FOLLOWING SCRIPT DON'T CHANGE ANYTHING AND JUST LAUNCH THE INSTANCE.**



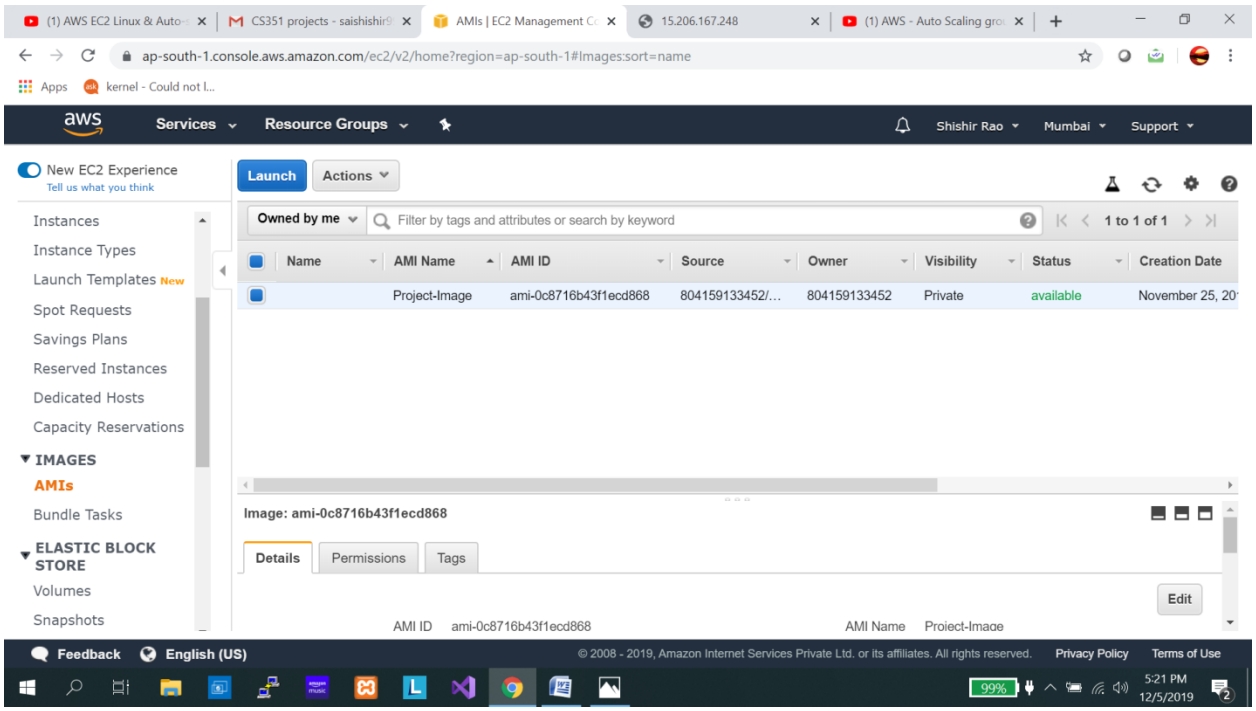
- **IN THE ABOVE PICTURE LOOK AT FIRST INSTANCE WE CAN SEE THE AUTOSCALING WEBSERVER.**



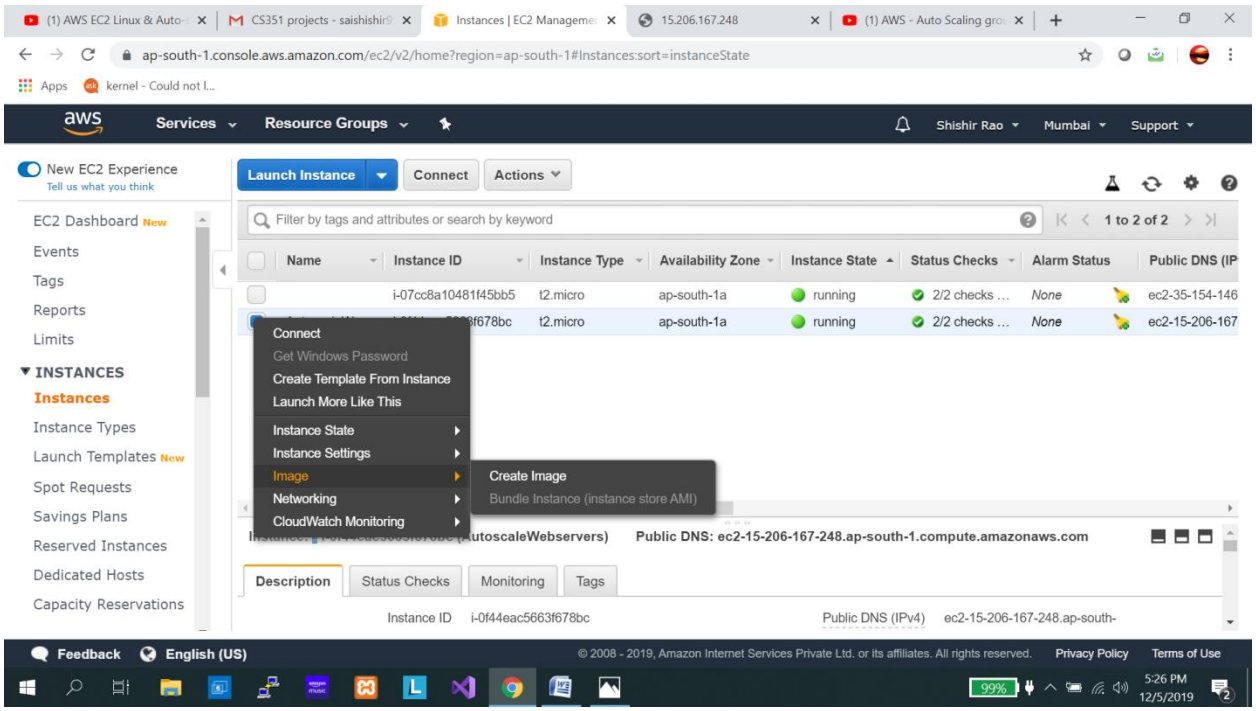
- **IN THIS PICTURE WE HAVE THE SAMPLE WEBSITE WHICH WE HAVE HOSTED.**
- **THE IP ADDRESS IN FIRST PICTURE AND THE SECOND ONE IS THE SAME.**

STEP 2:-

- **CREATE AN IMAGE OF THE ABOVE INSTANCE.**
- **WE CREATE AN IMAGE BECAUSE WE USE THIS IMAGE TO ACTUALLY CREATE OUR AUTOSCALING GROUP RATHER THAN A NEW SERVER.**



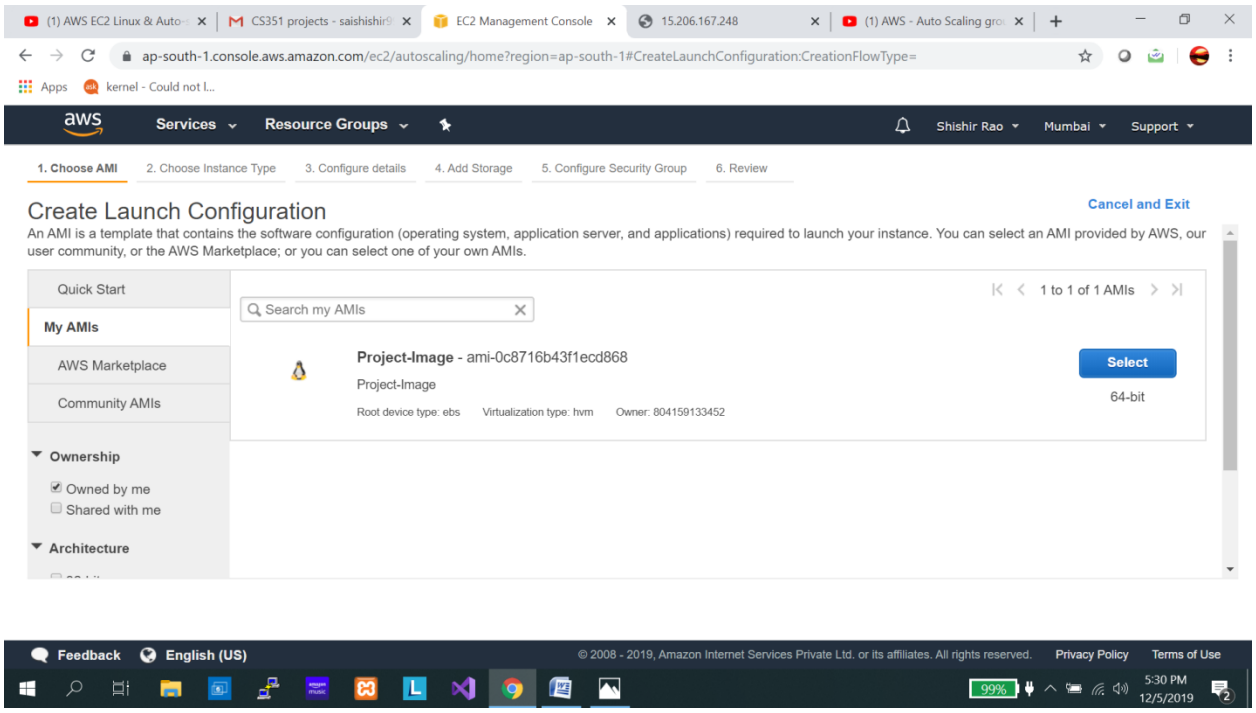
- **IMAGE OF MY WEBSERVER.**



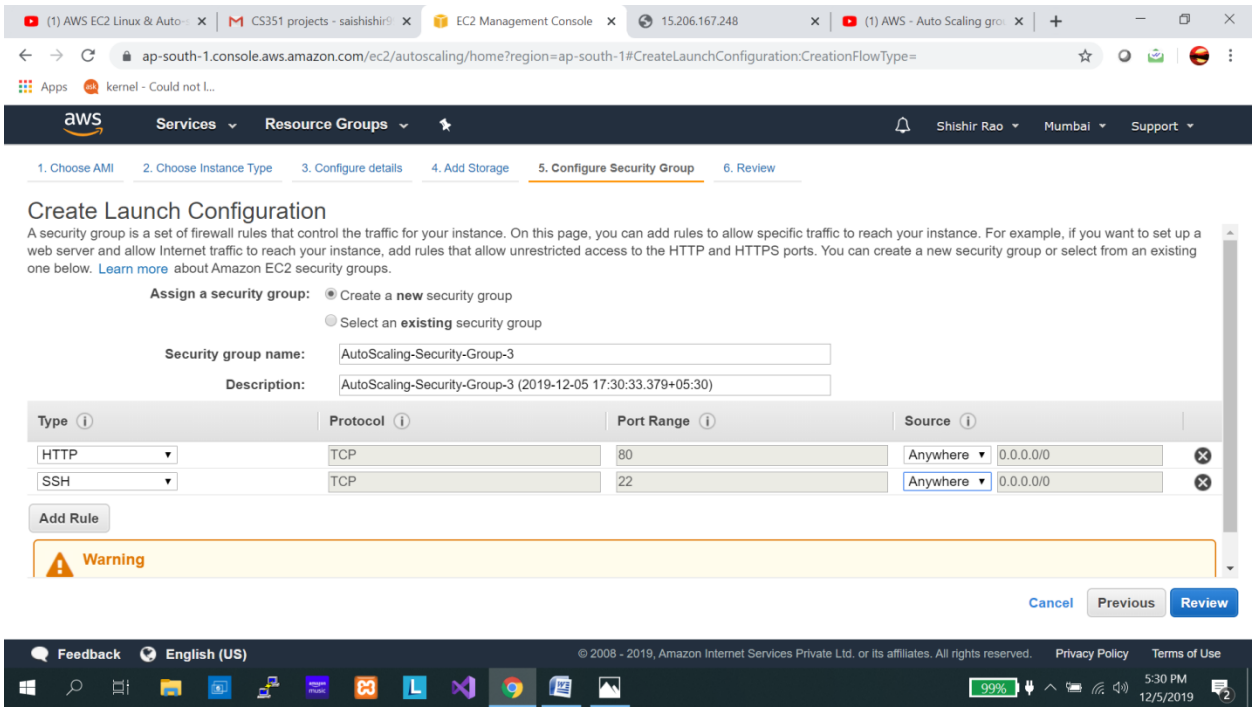
- **THIS IS HOW WE GRAPHICALLY CREATE AN IMAGE.**

STEP 3:-

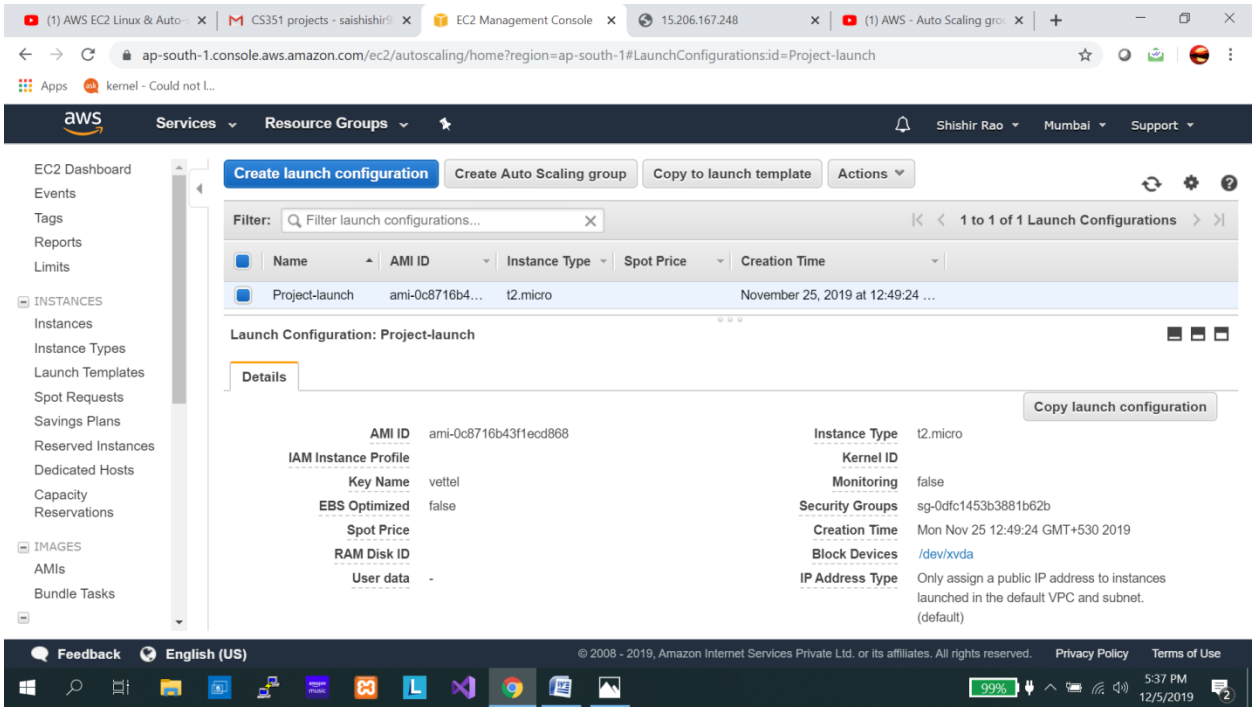
- **CREATE A LAUNCH CONFIGURATION.**
- **BEFORE CREATING AUTOSCALING GROUP WE NEED TO CREATE A LAUNCH CONFIGURATION.**
- **GO TO LAUNCH CONFIGURATION IN AWS AND SELECT CREATE LAUNCH CONFIGURATION AND SET IT TO T2.MICRO.**



- **SELECT THE AMI YOU HAVE CREATED RATHER THAN A LINUX SERVER.**
- **SELECT T2.MICRO IN TYPE AND ADD A NAME TO THE LAUNCH CONFIGURATION.**
- **DON'T CHANGE ANYTHING EXCEPT THE SECURITY SETTINGS.**



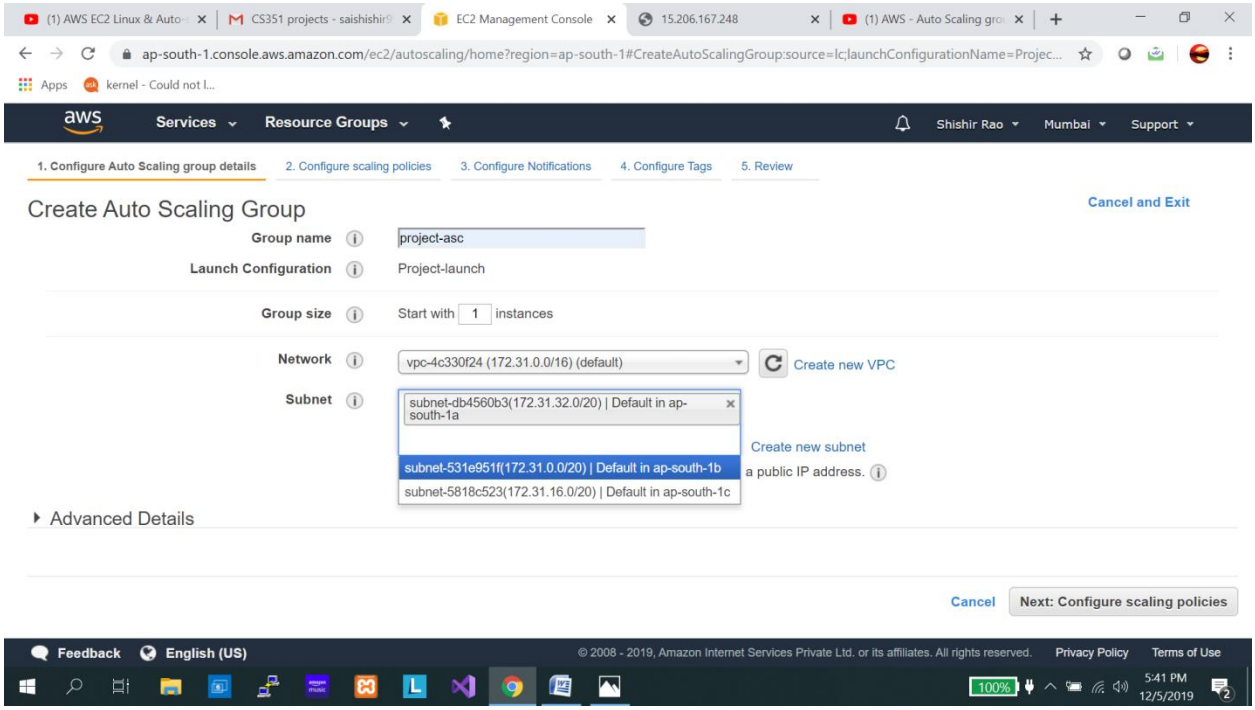
- **ADD THE ABOVE SECURITY SETTINGS AND CREATE THE LAUNCH CONFIGURATION.**



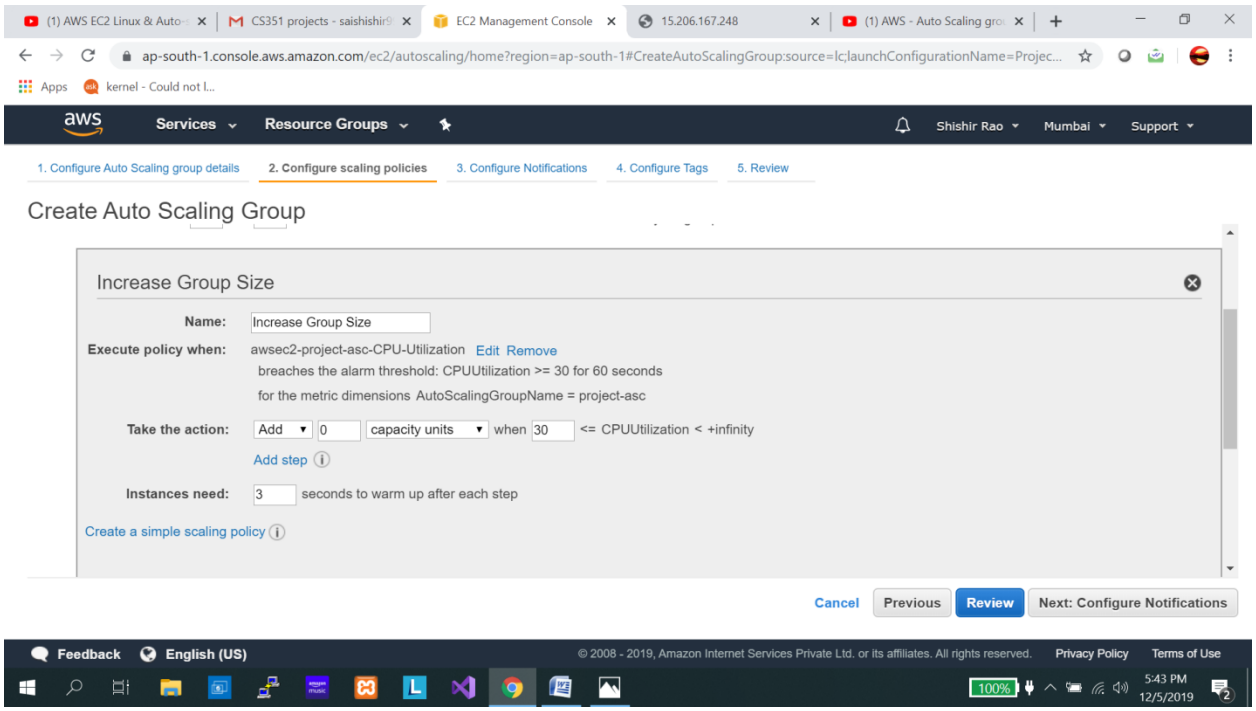
- **THIS IS MY LAUNCH CONFIGURATION.**

STEP 4:-

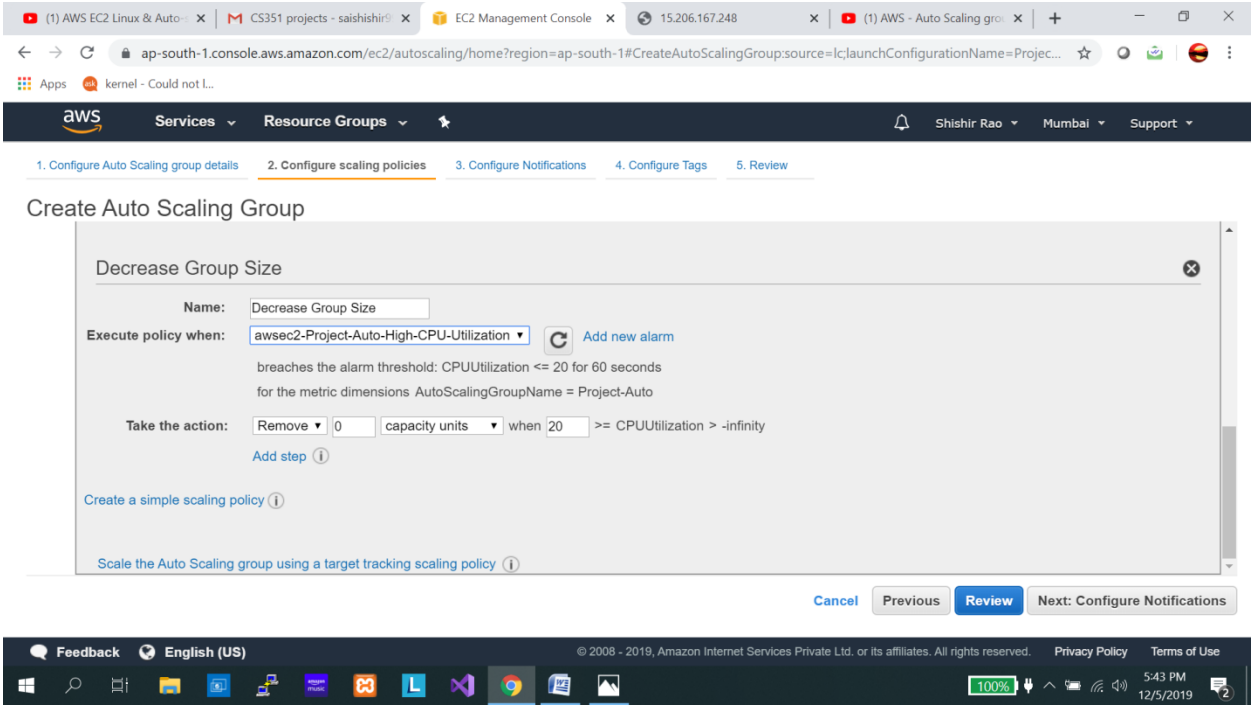
- **CREATE THE AUTOSCALING GROUP USING THE ABOVE LAUNCH CONFIGURATION.**
- **WE ONLY SET THE SCALING UP AND SCALING DOWN POLICIES IN THE AUTOSCALING GROUP AND SETUP NOTIFICATIONS.**



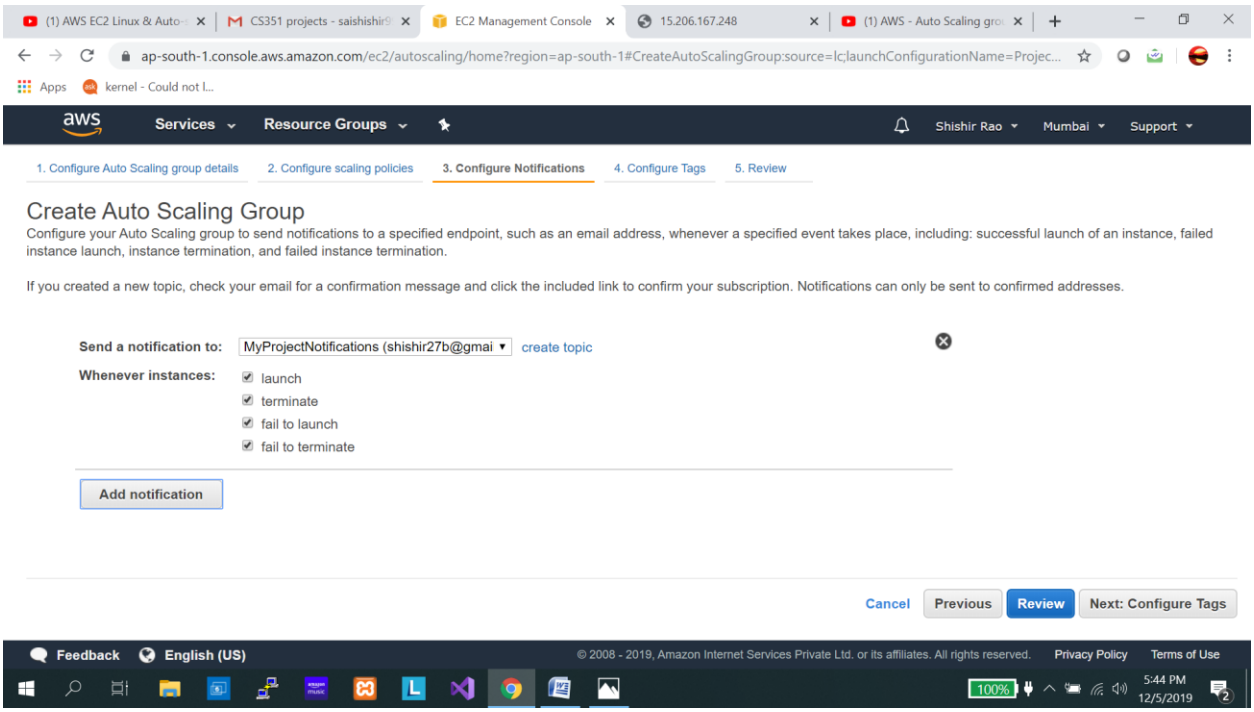
- **WHILE LAUNCHING WE SELECT THE SUBNETS IN WHICH OUR SERVER WILL WORK.(DONOT SELECT AP-SOUTH-1C, BECAUSE IT WILL NOT SUPPORT AUTOSCALING. ONLY SELECT 1A AND 1B)**
- **SET A GROUP NAME AND START WITH ANY NUMBER OF INSTANCES AS U WANT. I AM STARTING WITH ONE INSTANCE.**



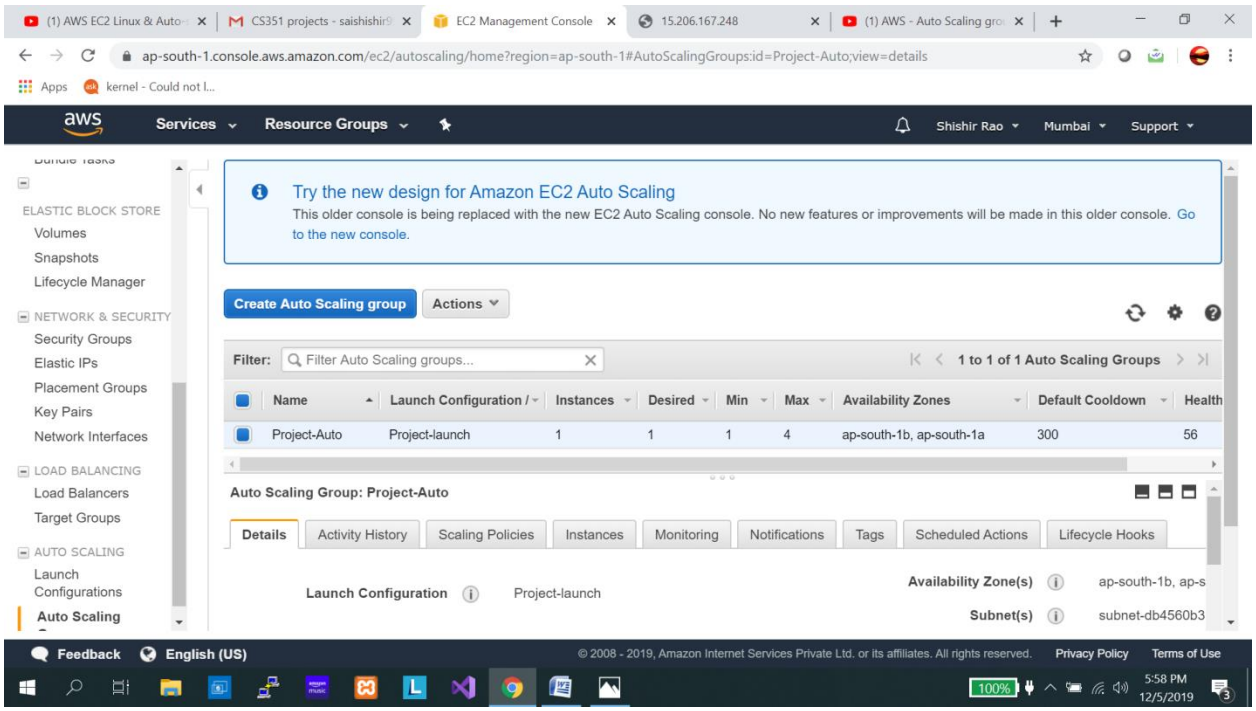
- **WE CREATE A SCALING UP POLICY HERE, THAT IS WE DECIDE AFTER WHAT PARAMETER WILL WE LAUNCH INSTANCES OR VIRTUAL MACHINES SO AS TO COMPLY WITH THE SYSTEM.**



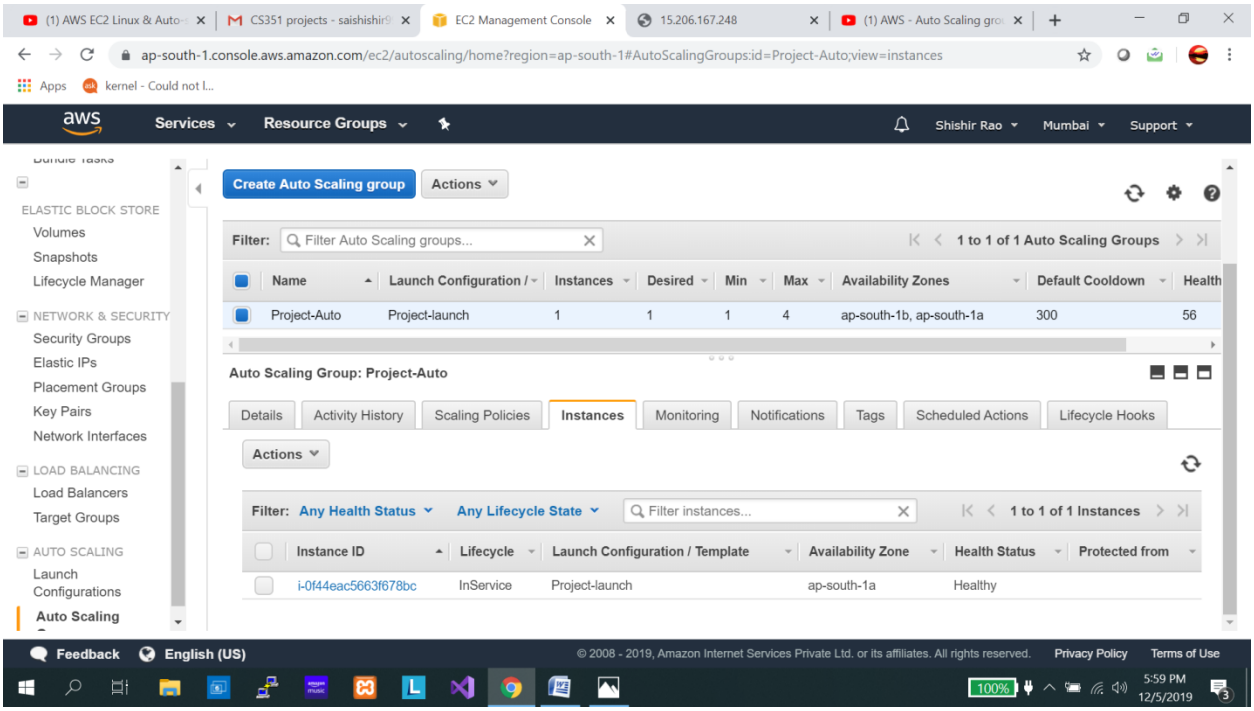
- **HERE WE HAVE CREATED THE SCALING DOWN POLICY SO AS TO WHEN WE SHOULD TERMINATE THE INSTANCES WHICH ARE NO MORE REQUIRED BECAUSE THE AVERAGE CPU UTILIZATION IS UNDER MARK AND OK.**
- **ONLY FOR DEMO PURPOSE WE HAVE SET THE AVG UTILIZATION AS 30 FOR SCALE UP AND 20 FOR SCALE DOWN.**
- **WE ALSO PUT THE TIME AS 1 MINUTE BECAUSE THIS IS A DEMO AND WE DON'T WANT TO WASTE MUCH TIME.**



- **WE THE SET THE NOTIFICATIONS WITH EMAIL, SO THAT EVERYTIME A SERVER IS LAUNCHED OR TERMINATED WE GET A NOTIFICATION FROM AWS.**
- **LAUNCH THE AUTOSCALING GROUP.**



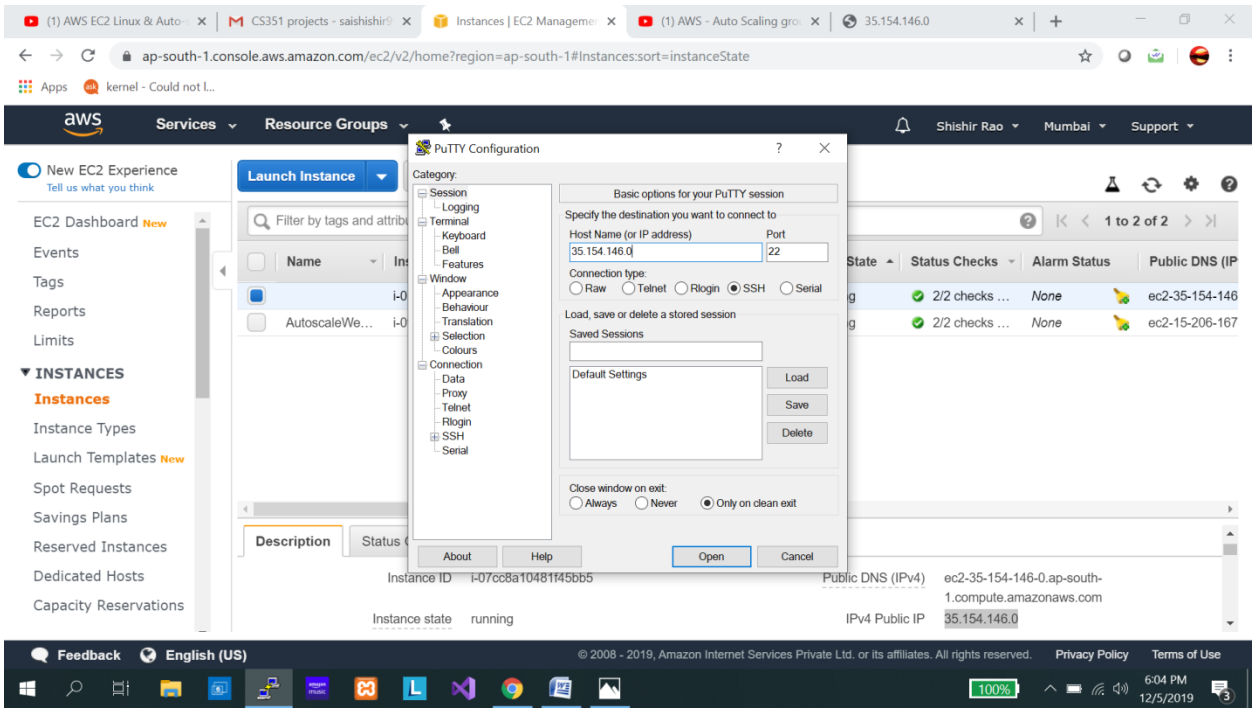
- **AUTOSCALING GROUP HAS BEEN LAUNCHED**



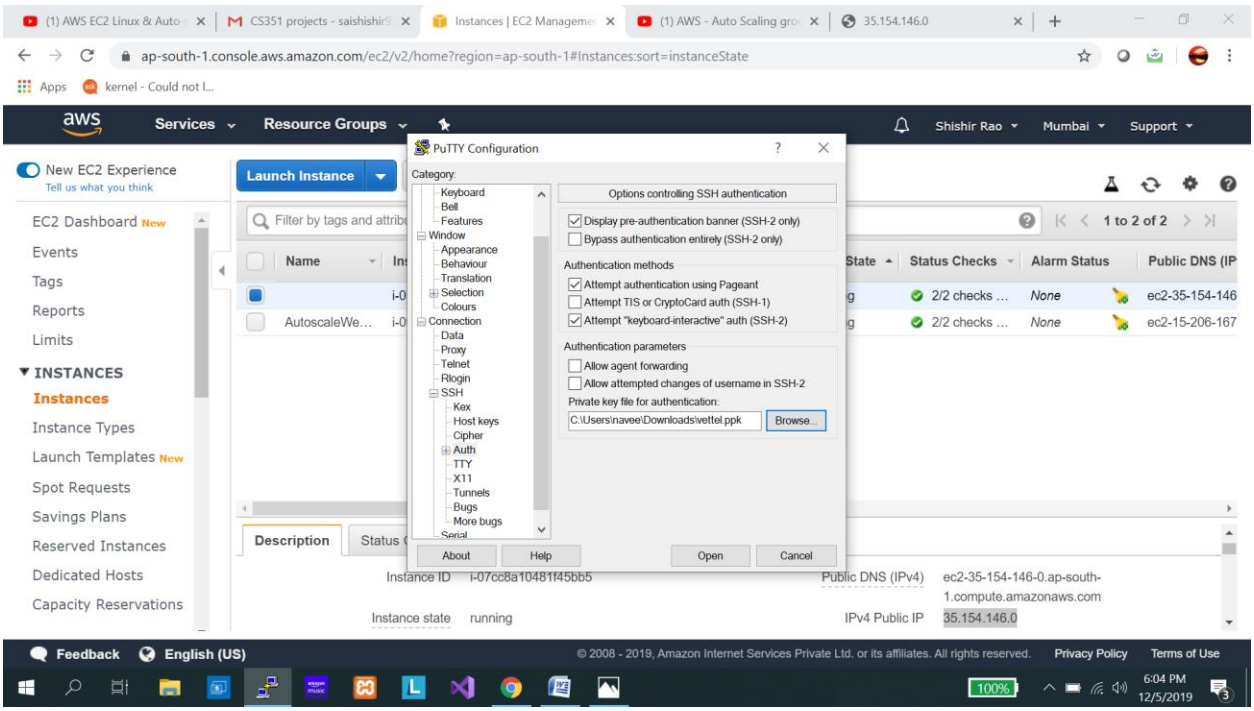
- **ONE INSTANCE HAS ALREADY BEEN LAUNCHED BECAUSE WE HAVE MENTIONED IT ALREADY THAT WE WILL START WITH ONE INSTANCE.**

STEP 5:-

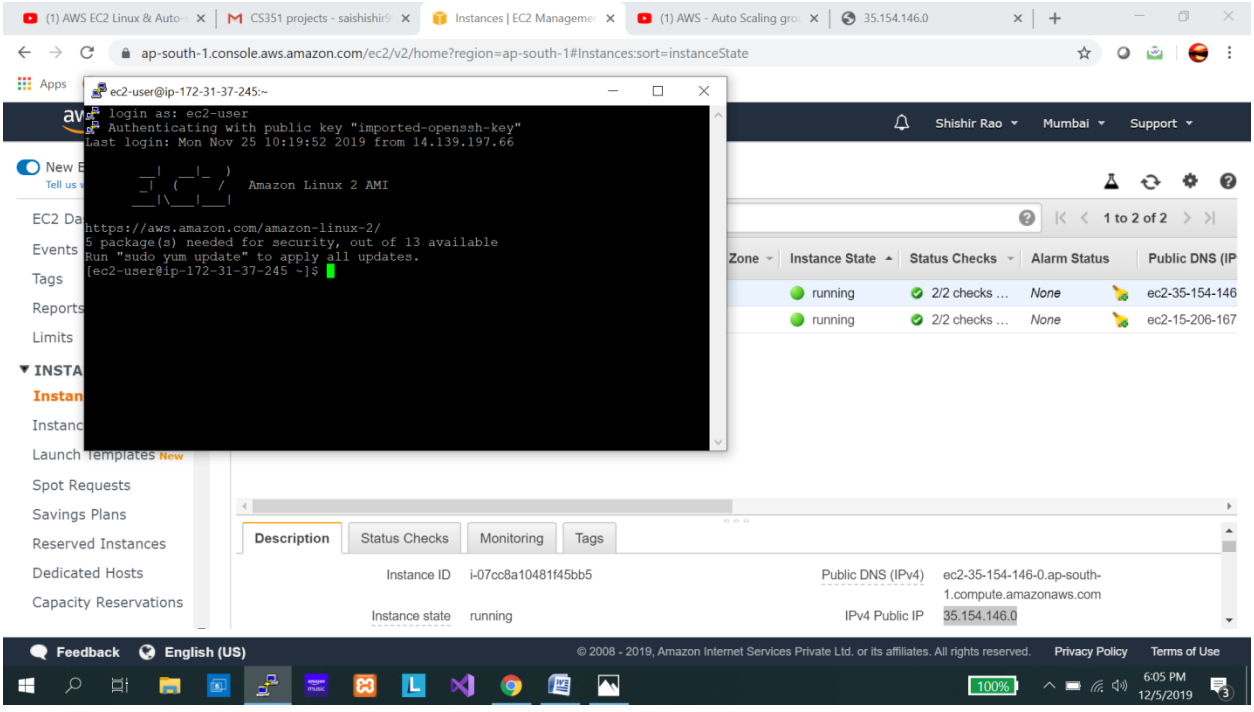
- **USE YOUR SSH KEY TO LOGIN INTO THE SERVER USING PPK AND PEM FILES.**



- **USE PUTTY AND SET THE IP ADDRESS TO THE ONE IN YOUR WEBSERVER INSTANCE.**



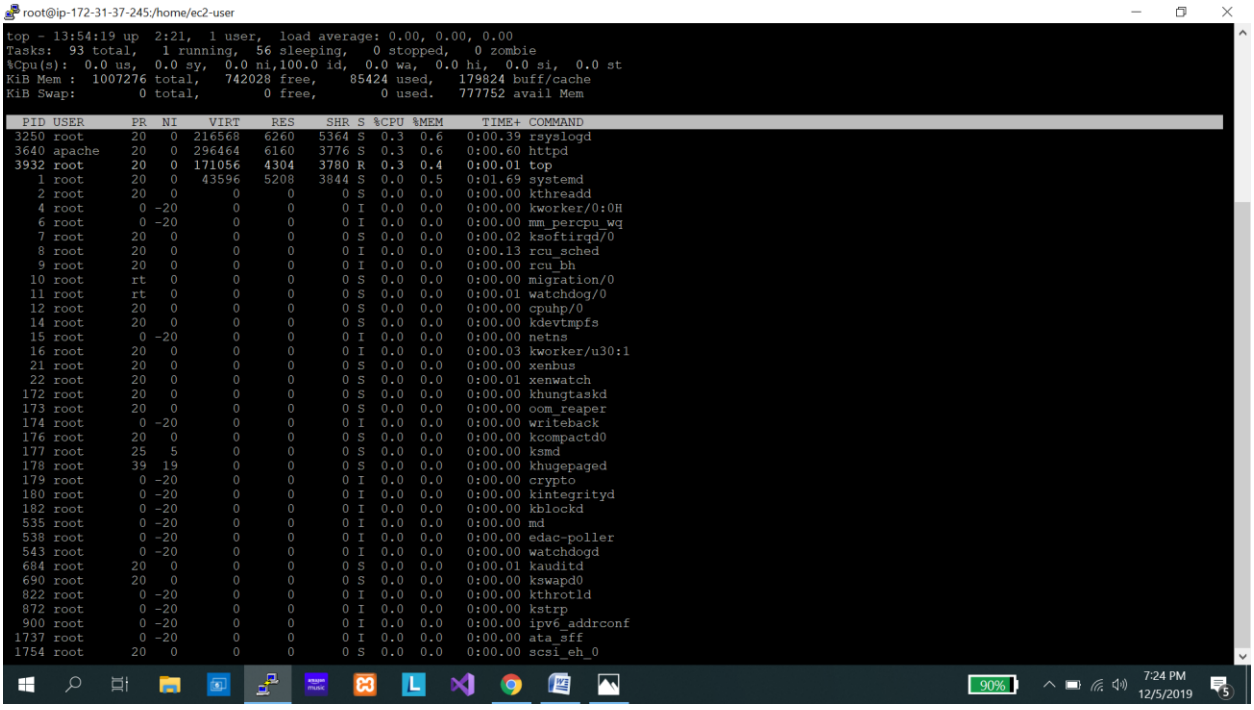
- **WE ALSO PUT THE PPK FILE WHICH WE HAVE CREATED FROM PEM FILE.**
- **WE THEN CLICK OPEN.**



- **LOGIN AS “ec2-user”.**

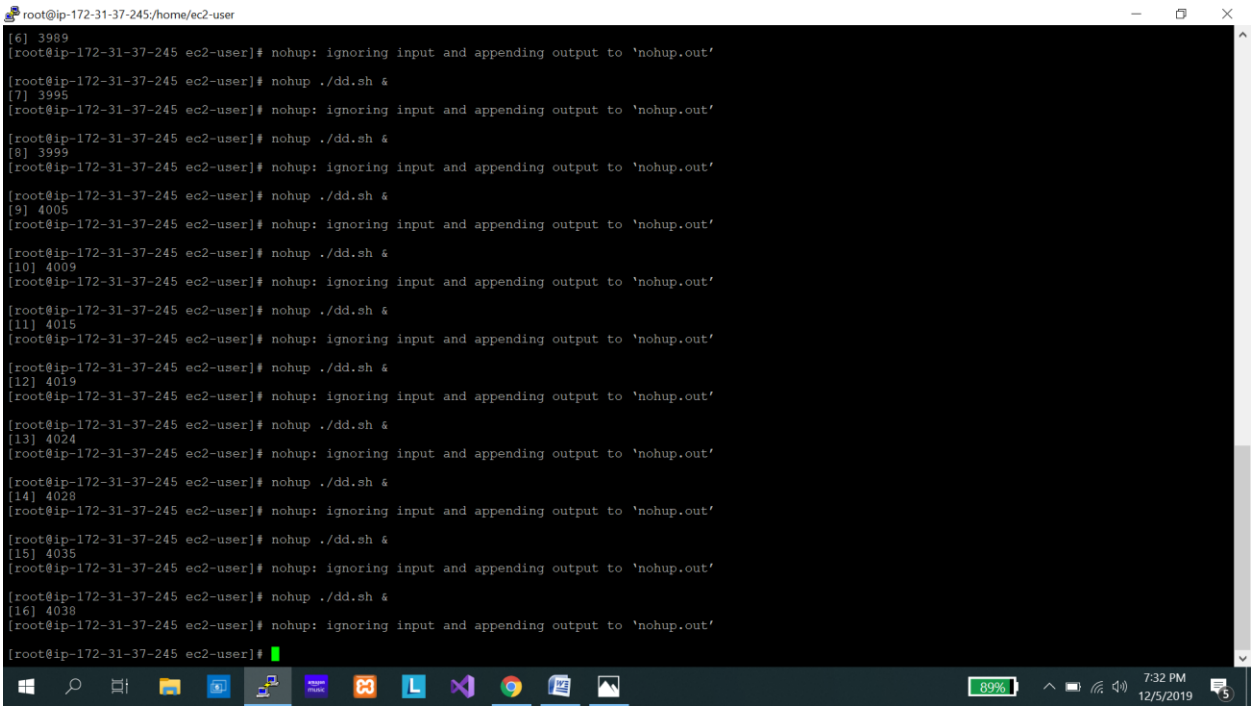
STEP 6:-

- **INCREASE THE LOAD ON WEBSERVER.**
- **WE USE THE FOLLOWING SCRIPT TO DO IT SO.**
- **WE CREATE A FILE WITH NAME INCREASE.SH**
- **dd if=/var/zero of=/dev/null bs=50000 count=10000**
- **PUT THE ABOVE LINE INTO THE FILE AND THEN RUN IT UNTILL THE LOAD INCREASES.**
- **WE USE THE “TOP” COMMAND TO FIND OUT THE CPU UTILIZATION.**

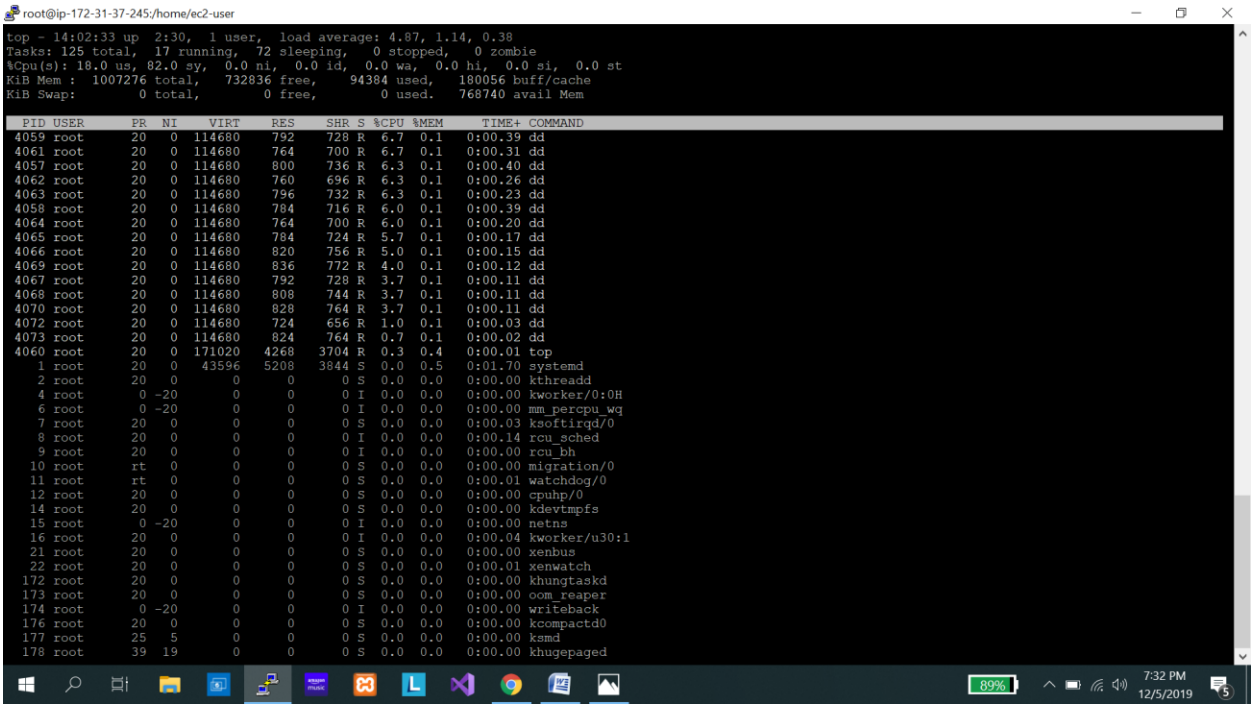


- **WE CAN SEE THAT THE CPU UITLIZATION IS ZERO WHEN THERE IS LOAD ON THE SERVER.**
- **NOW CHANGE THE SCRIPT TO EXECUTABLE FORM AND RUN IT.**

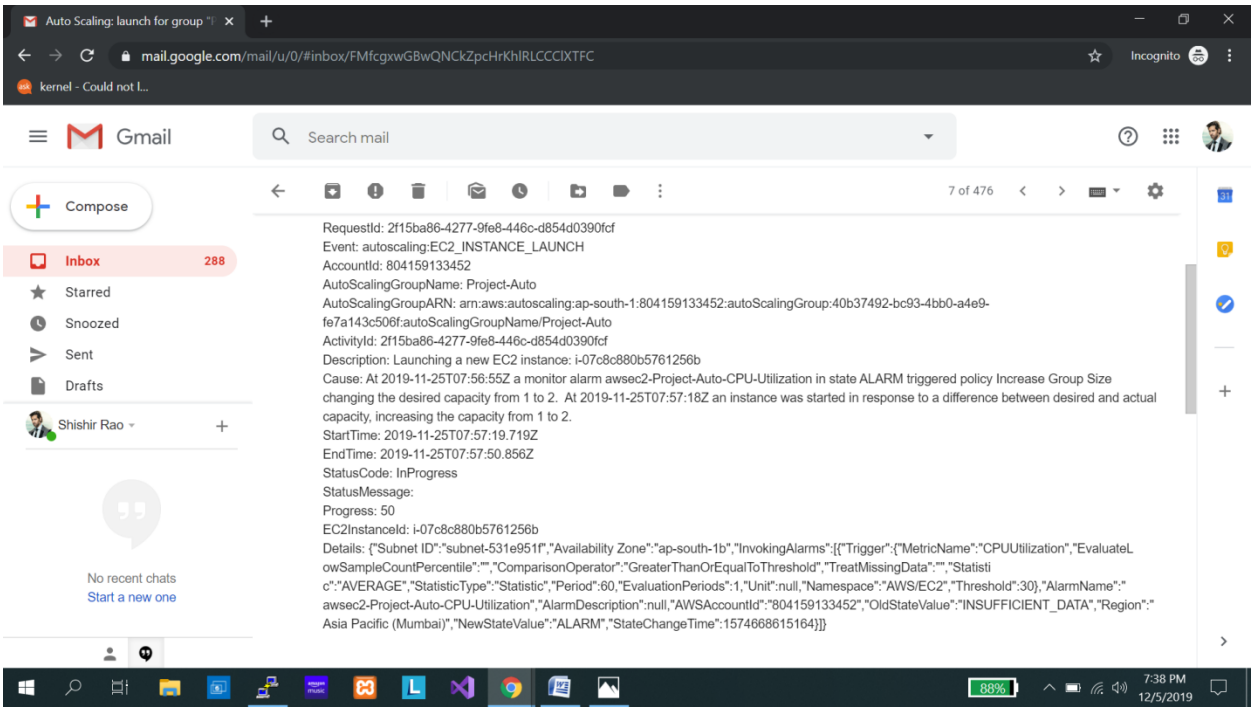
- **CHMOD 755 INCREASE.SH**
- **THE COMMAND TO EXECUTE IS “nohup ./increase.sh &”**
- **EXECUTE THE COMMAND AS MANY TIMES AS YOU CAN SO AS TO INCREASE THE LOAD FROM 0 TO SOMETHING MORE THAN 30.**



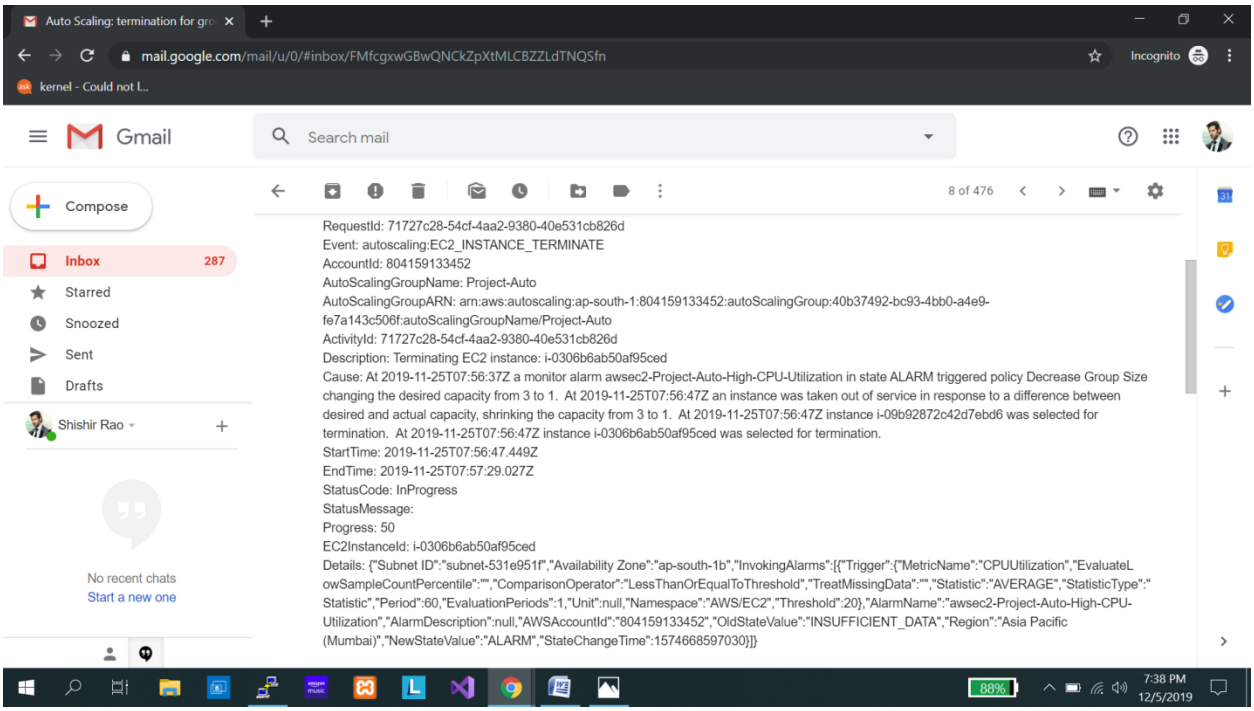
- **RUNNING THE COMMAND SO MANY TIMES.**



- **WE CAN SEE THAT THE CPU UTILIZATION HAS GONE UP FROM 0 TO 18.**
- **EXECUTE UNTILL IT CROSSES 30 AND STAYS THERE FOR ATLEAST A MINUTE.**
- **ONCE IT CROSSES 30 AND STAYS THERE FOR 1 MINUTE YOU WILL RECEIVE NOTIFICATION TO YOUR MAIL ADDRESS.**



- **MAIL I RECEIVED REAGARDING THE INSTANCE LAUNCH.**
- **ONCE THE CPU COMES BELOW 20 AND STAYS FOR ONE MINUTE YOU WILL RECEIVE ANOTHER NOTIFICATION REGARDING INSTANCE TERMINATION.**



- **NOTIFICATION WE RECEIVE REGARDING INSTANCE TERMINATION.**

CONCLUSION:-

- **WE HAVE FINALLY SIMULATED THE ENTIRE HYPERVISOR BY CONSIDERING THE ONLINE SERVER AS OUR SYSTEM AND INSTANCES AS VIRTUAL MACHINES.**
- **WE ALSO MADE IT INTELLIGENT BECAUSE WE NEVER LAUNCHED AN INSTANCE OR TERMINATED ANY WE JUST INCREASED THE CPU UTILIZATION AND IT AUTOMATICALLY LAUNCHED BY ITSELF.**