Cloud Computing Final Project Report

Group 4

Vigneshwar Muriki

Mohit Battu

Lokesh Kola

Srikar Daruru

Problem Description

The final project is to deploy a flask application into Amazon Web Services (AWS) and apply autoscaling to satisfy the requirements scalability with respect to computation and high availability of the computation.

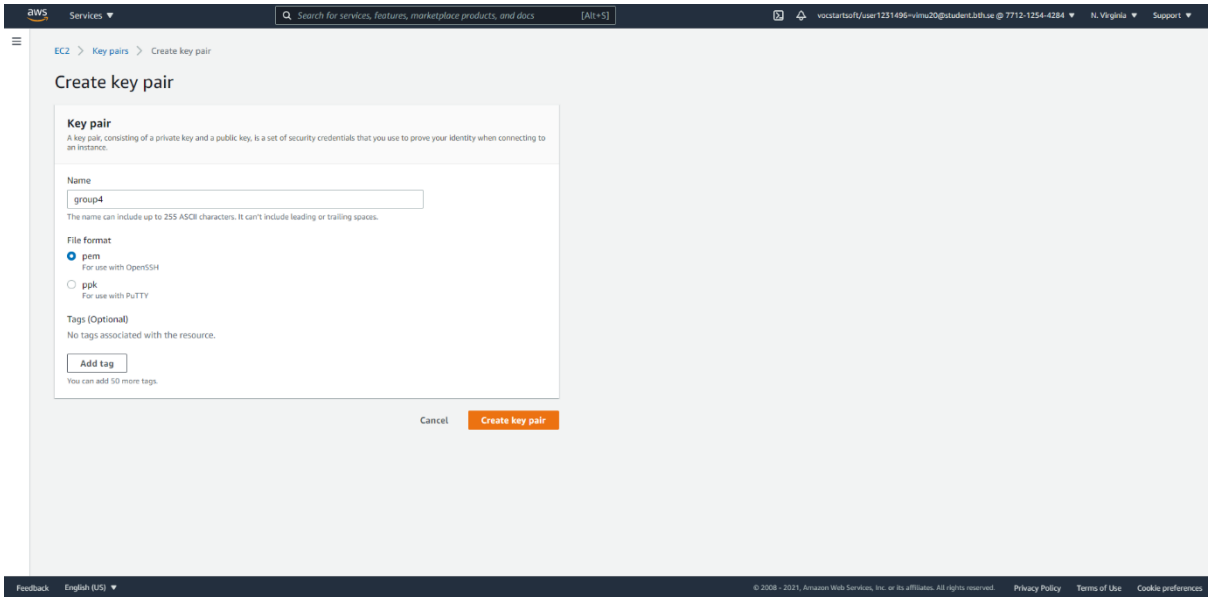
Design Description

In this project, we selected to deploy flask application into AWS. To do this, an ubuntu EC2 instance AMI is launched and connected to the server. To deploy flask application into ubuntu server, Nginx and Gunicorn3 WSGI are configured. When the flask application is deployed into AWS, auto-scaling was performed by adding required scaling policies, which configures the flask application average CPU use and implement scaling policies.

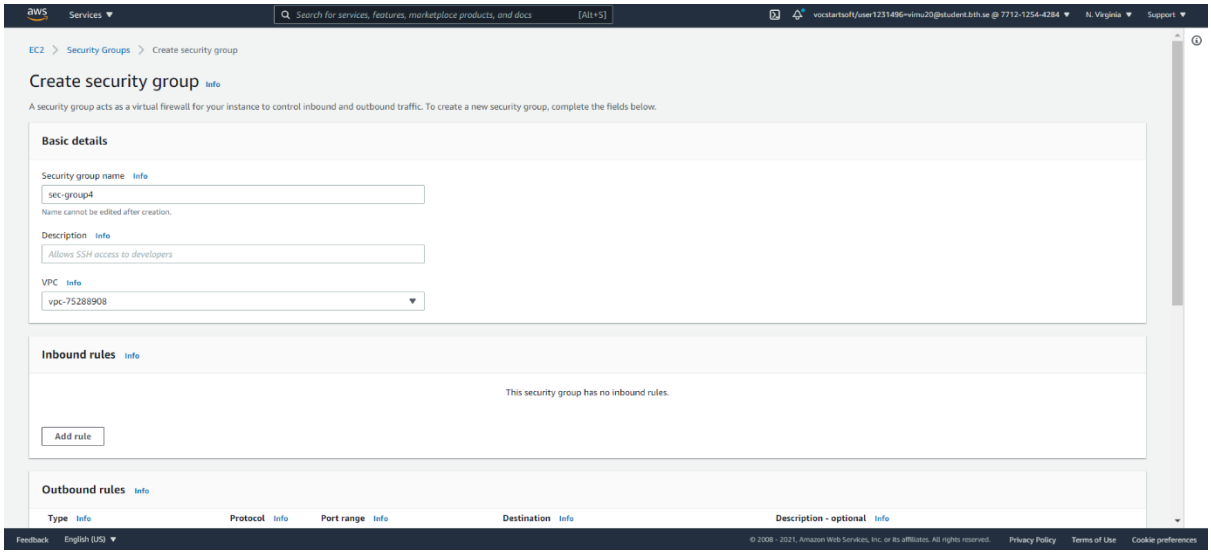
Implementation

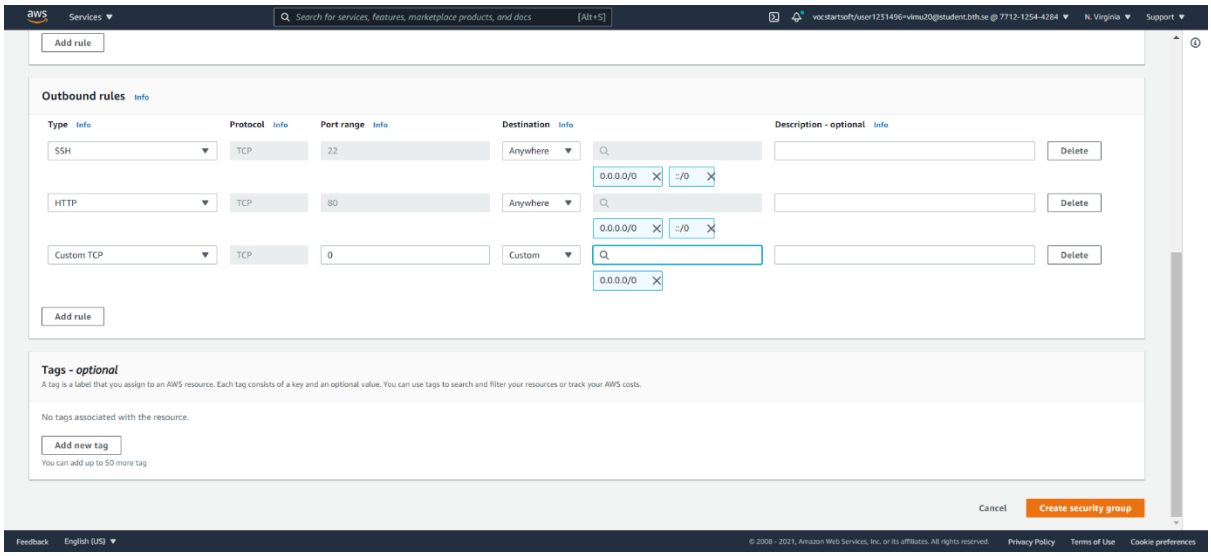
The first step to launch an EC2 instance is to create a key pair and a security group.

The key pair is created and saved as pem file type.



A security group is created with necessary inbound and outbound rules.





Select EC2 option in the AWS services.

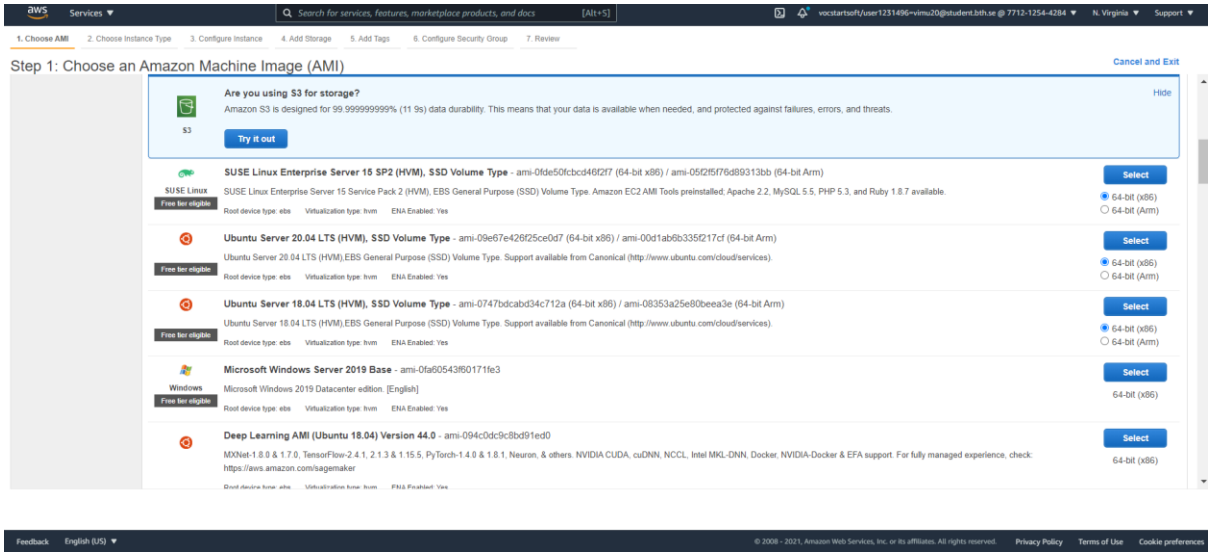
To launch an instance, we need to choose Amazon Machine Image (AMI). We have selected ubuntu server image which is available for free.

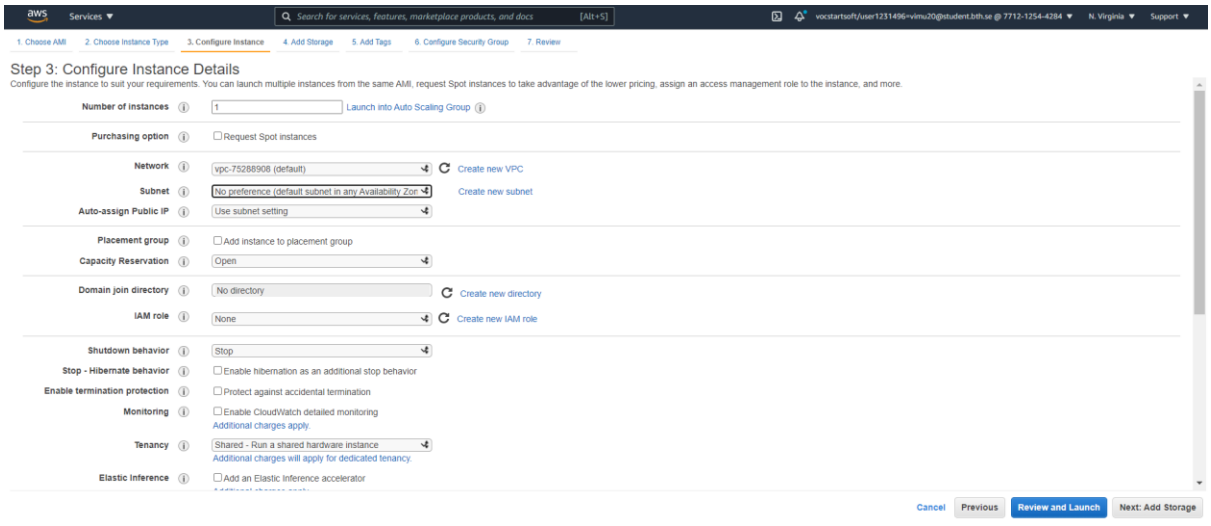
While configuring the instance details, we have selected default vpc and subnets provided by Aws.

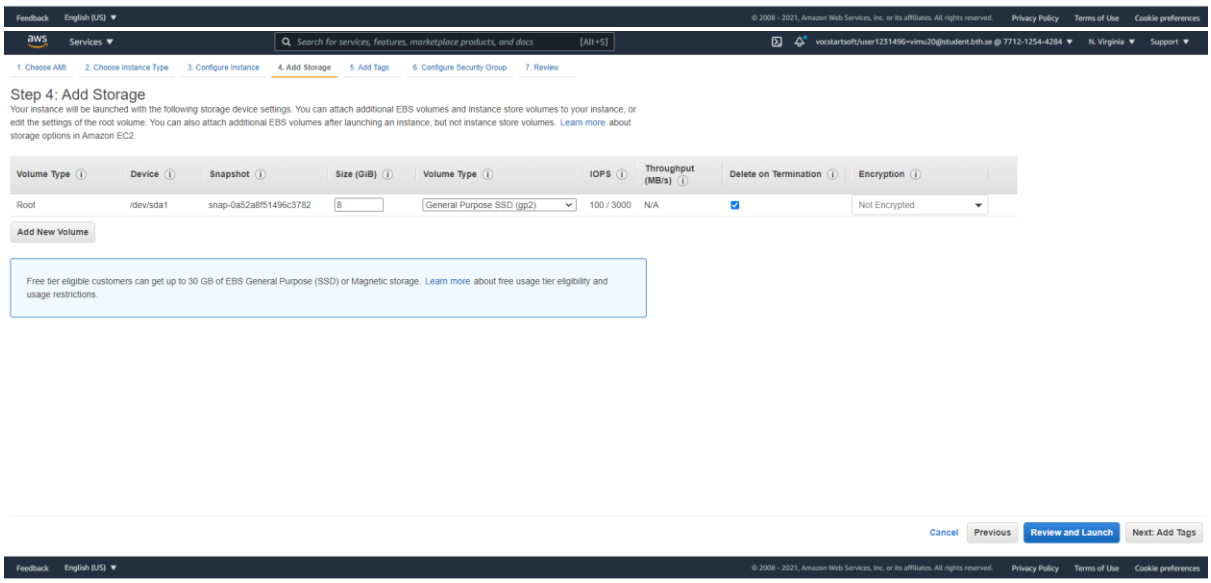
We add storage which is provided default.

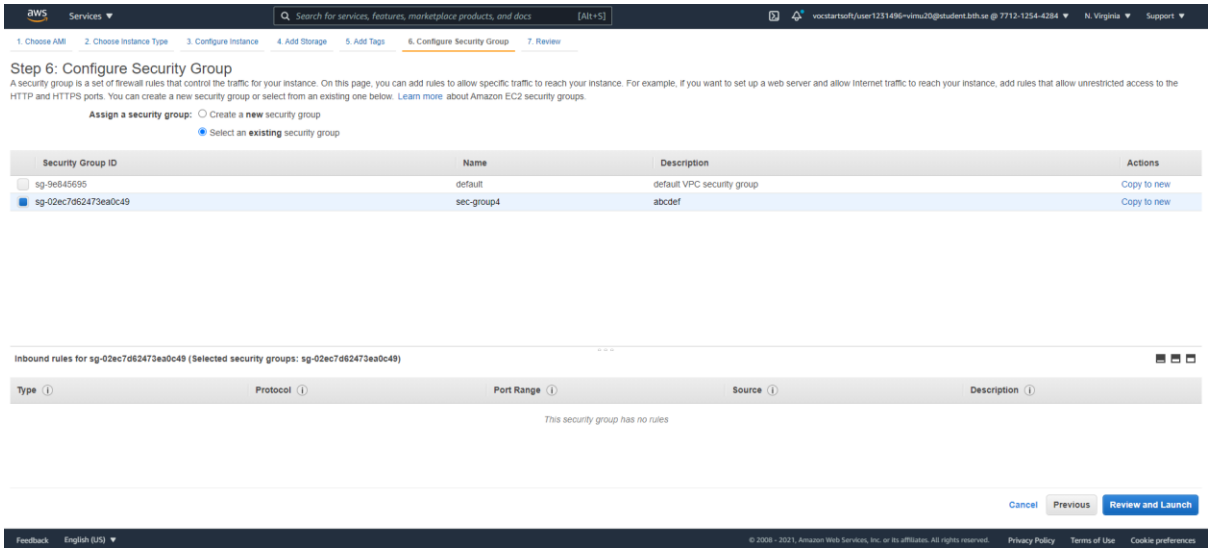
Then we have selected security group which was created previous and selected suitable inbound and outbound rules for the instance.

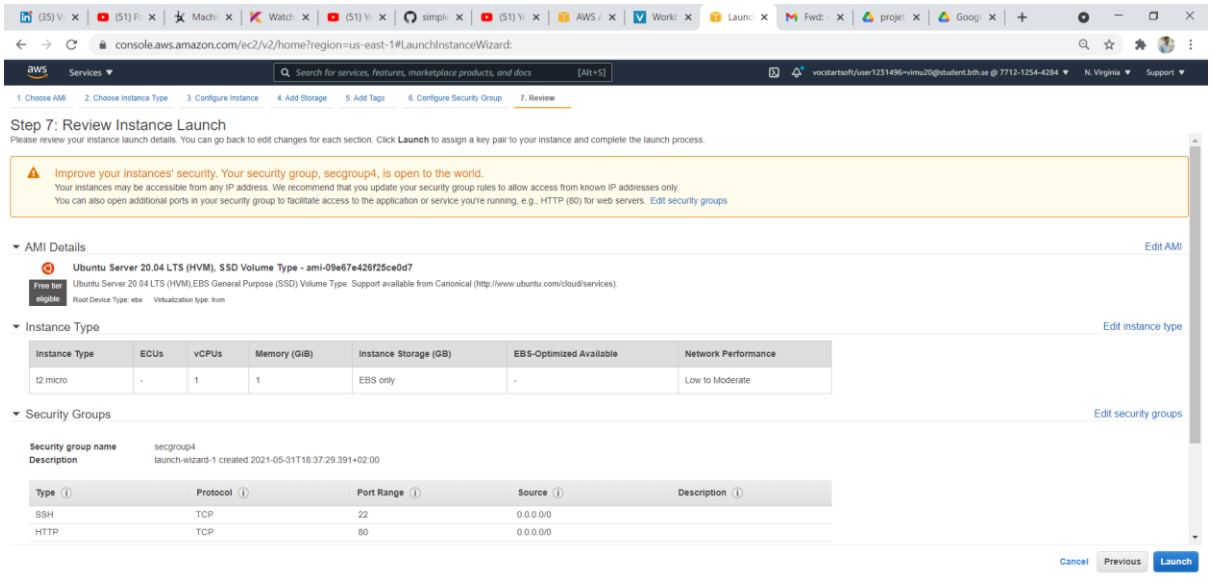
In the last step of launching an instance we have to select the key pair which needs to associated to the instance. Here we have selected the key pair which was created previously.

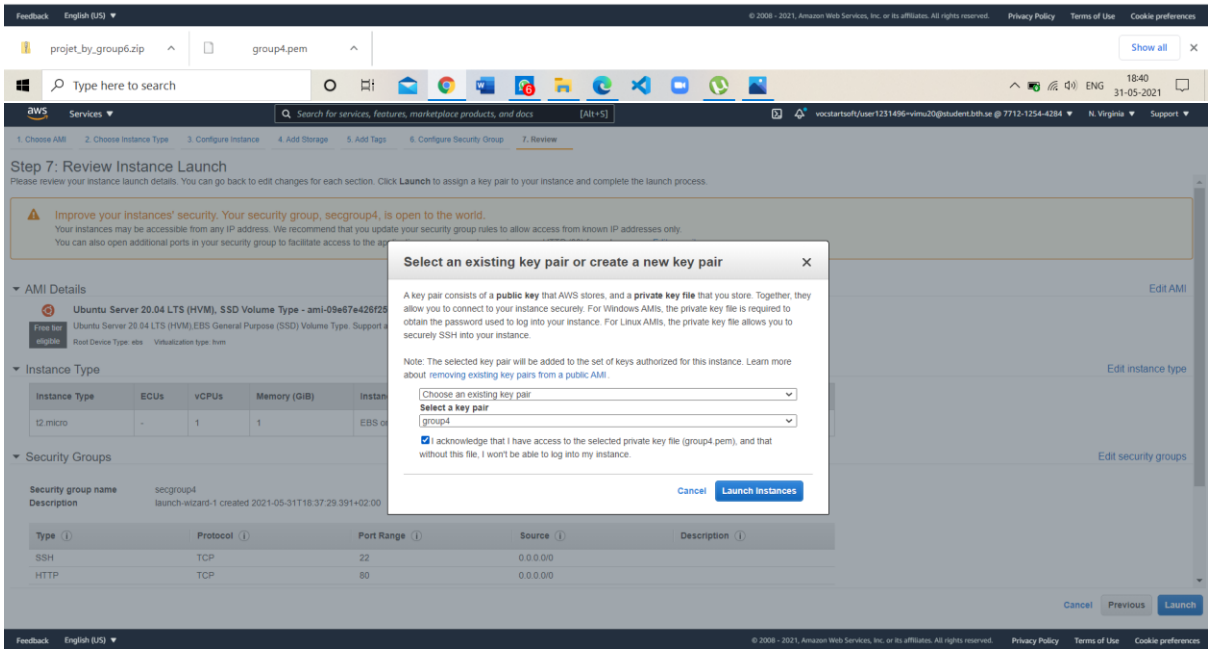

Services
Search for services, features, marketplace products, and docs [Alt+S]
vocstartsoft/user1231496=vimu20@student.bth.se @ 7712-1254-4284
N. Virginia
Support
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
Step 1: Choose an Amazon Machine Image (AMI)
Cancel and Exit
Are you using S3 for storage?
Hide
Amazon S3 is designed for 99.999999999% (11 9s) data durability. This means that your data is available when needed, and protected against failures, errors, and threats.
Try it out
SUSE Linux Enterprise Server 15 SP2 (HVM), SSD Volume Type - ami-0fde50fcbcd46f2f7 (64-bit x86) / ami-05f2f5f76d89313bb (64-bit Arm)
SUSE Linux Enterprise Server 15 Service Pack 2 (HVM), EBS General Purpose (SSD) Volume Type. Amazon EC2 AMI Tools preinstalled; Apache 2.2, MySQL 5.5, PHP 5.3, and Ruby 1.8.7 available.
Free tier eligible
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
Select
64-bit (x86)
64-bit (Arm)
Ubuntu Server 20.04 LTS (HVM), SSD Volume Type - ami-09e67e426f25ce0d7 (64-bit x86) / ami-00d1ab6b335f217cf (64-bit Arm)
Ubuntu Server 20.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).
Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-0747bdcabd34c712a (64-bit x86) / ami-08353a25e80beea3e (64-bit Arm)
Ubuntu Server 18.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).
Microsoft Windows Server 2019 Base - ami-0fa60543f60171fe3
Microsoft Windows 2019 Datacenter edition. [English]
Deep Learning AMI (Ubuntu 18.04) Version 44.0 - ami-094c0dc9c8bd91ed0
MXNet-1.8.0 & 1.7.0, TensorFlow-2.4.1, 2.1.3 & 1.15.5, PyTorch-1.4.0 & 1.8.1, Neuron, & others. NVIDIA CUDA, cuDNN, NCCL, Intel MKL-DNN, Docker, NVIDIA-Docker & EFA support. For fully managed experience, check: https://aws.amazon.com/sagemaker
Feedback
English (US)
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Cookie preferences


Services
Search for services, features, marketplace products, and docs [Alt+S]
vocstartsoft/user1231496=vimu20@student.bth.se @ 7712-1254-4284
N. Virginia
Support
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
Step 3: Configure Instance Details
Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.
Number of instances
1
Launch into Auto Scaling Group
Purchasing option
Request Spot instances
Network
vpc-75288908 (default)
Create new VPC
Subnet
No preference (default subnet in any Availability Zon
Create new subnet
Auto-assign Public IP
Use subnet setting
Placement group
Add instance to placement group
Capacity Reservation
Open
Domain join directory
No directory
Create new directory
IAM role
None
Create new IAM role
Shutdown behavior
Stop
Stop - Hibernate behavior
Enable hibernation as an additional stop behavior
Enable termination protection
Protect against accidental termination
Monitoring
Enable CloudWatch detailed monitoring
Additional charges apply.
Tenancy
Shared - Run a shared hardware instance
Additional charges will apply for dedicated tenancy.
Elastic Inference
Add an Elastic Inference accelerator
Cancel
Previous
Review and Launch
Next: Add Storage
Feedback
English (US)
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Cookie preferences

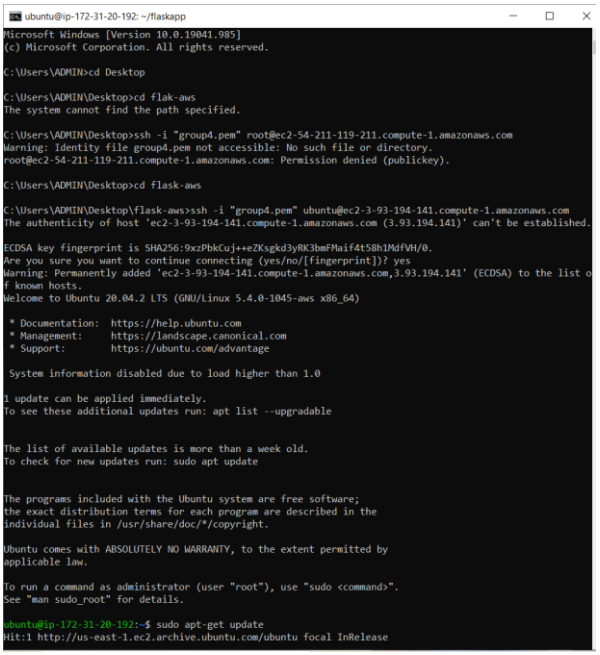

Services
Search for services, features, marketplace products, and docs [Alt+S]
vocstartsoft/user1231496=vimu20@student.bth.se @ 7712-1254-4284
N. Virginia
Support
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
Step 4: Add Storage
Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. Learn more about storage options in Amazon EC2.
Volume Type
Device
Snapshot
Size (GiB)
Volume Type
IOPS
Throughput (MB/s)
Delete on Termination
Encryption
Root
/dev/sda1
snap-0a52a8f51496c3782
8
General Purpose SSD (gp2)
100 / 3000
N/A
Not Encrypted
Add New Volume
Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. Learn more about free usage tier eligibility and usage restrictions.
Cancel
Previous
Review and Launch
Next: Add Tags
Feedback
English (US)
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Cookie preferences


Step 6: Configure Security Group
A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.
Assign a security group: Create a new security group
Select an existing security group
Security Group ID
Name
Description
Actions
sg-9e845695
default
default VPC security group
Copy to new
sg-02ec7d62473ea0c49
sec-group4
abcdef
Copy to new
Inbound rules for sg-02ec7d62473ea0c49 (Selected security groups: sg-02ec7d62473ea0c49)
Type
Protocol
Port Range
Source
Description
This security group has no rules
Cancel
Previous
Review and Launch


console.aws.amazon.com/ec2/v2/home?region=us-east-1#LaunchInstanceWizard:
Step 7: Review Instance Launch
Please review your instance launch details. You can go back to edit changes for each section. Click Launch to assign a key pair to your instance and complete the launch process.
Improve your instances' security. Your security group, secgroup4, is open to the world.
Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only.
You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. Edit security groups
AMI Details
Edit AMI
Ubuntu Server 20.04 LTS (HVM), SSD Volume Type - ami-09e67e426f25ce0d7
Free tier eligible
Ubuntu Server 20.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).
Root Device Type: ebs
Virtualization type: hvm
Instance Type
Edit instance type
Instance Type
ECUs
vCPUs
Memory (GiB)
Instance Storage (GB)
EBS-Optimized Available
Network Performance
t2.micro
-
1
1
EBS only
-
Low to Moderate
Security Groups
Edit security groups
Security group name
secgroup4
Description
launch-wizard-1 created 2021-05-31T18:37:29.391+02:00
Type
Protocol
Port Range
Source
Description
SSH
TCP
22
0.0.0.0/0
HTTP
TCP
80
0.0.0.0/0
Cancel
Previous
Launch


projet_by_group6.zip
group4.pem
Show all
Type here to search
ENG
18:40
31-05-2021
Select an existing key pair or create a new key pair
A key pair consists of a public key that AWS stores, and a private key file that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.
Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.
Choose an existing key pair
Select a key pair
group4
I acknowledge that I have access to the selected private key file (group4.pem), and that without this file, I won't be able to log into my instance.
Cancel
Launch Instances

After launching the instance, we can connect the instance to the ubuntu server using either putty or using command prompt. We have connected the instance to the ubuntu server using the below command in the command prompt.



After connecting the instance to the ubuntu server, following commands are used in the ubuntu server.

sudo apt-get update

The above command does not install new versions. Instead it updates the list of packages for further upgrades that needs upgrading and also the packages which newly entered into repositories.

sudo apt-get upgrade

Using this above command diaplsys the list of updates which are available, we click y if we need all to be upgraded.

Sudo apt-get install python3

Using this command it installs python latest version into ubuntu server.

sudo apt-get install python3-pip

The above command installs pip is installed for python3

whereis pip3

This command gives the location where pip was installed.

Sudo pip3 install flask

Since we are deploying flask application, flask is downloaded using the above command.

sudo apt-get install nginx

The reason for using nginx is that it provides load balancing support, compatibility, websites are made faster such that traffic maintained, and easy to configure.

sudo apt-get install gunicorn

Gunicorn is a Web Server Gateway Interface (WSGI) application server. It is used to make web servers and python web apps communicate each other.

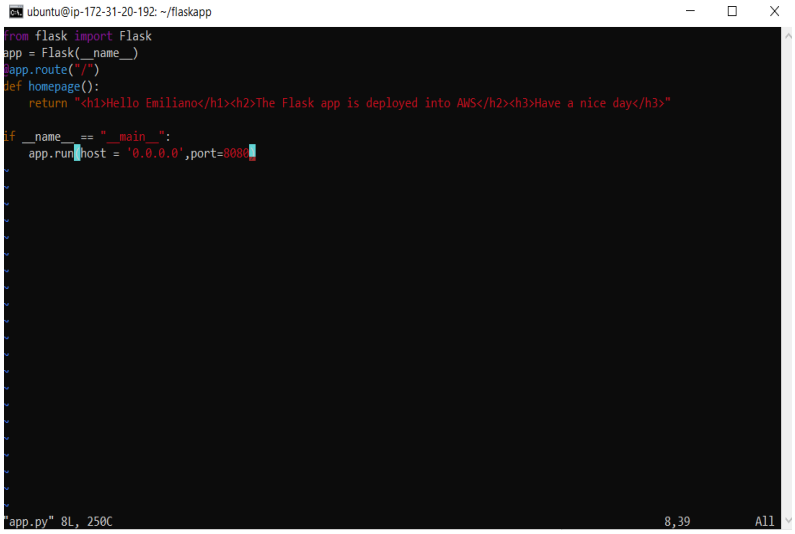
mkdir flaskapp

Inside the ubuntu server a new directory is created with name as flaskapp.

cd flaskapp

we changed the flaskapp directory where our flask app is created using the below command.

vi app.py



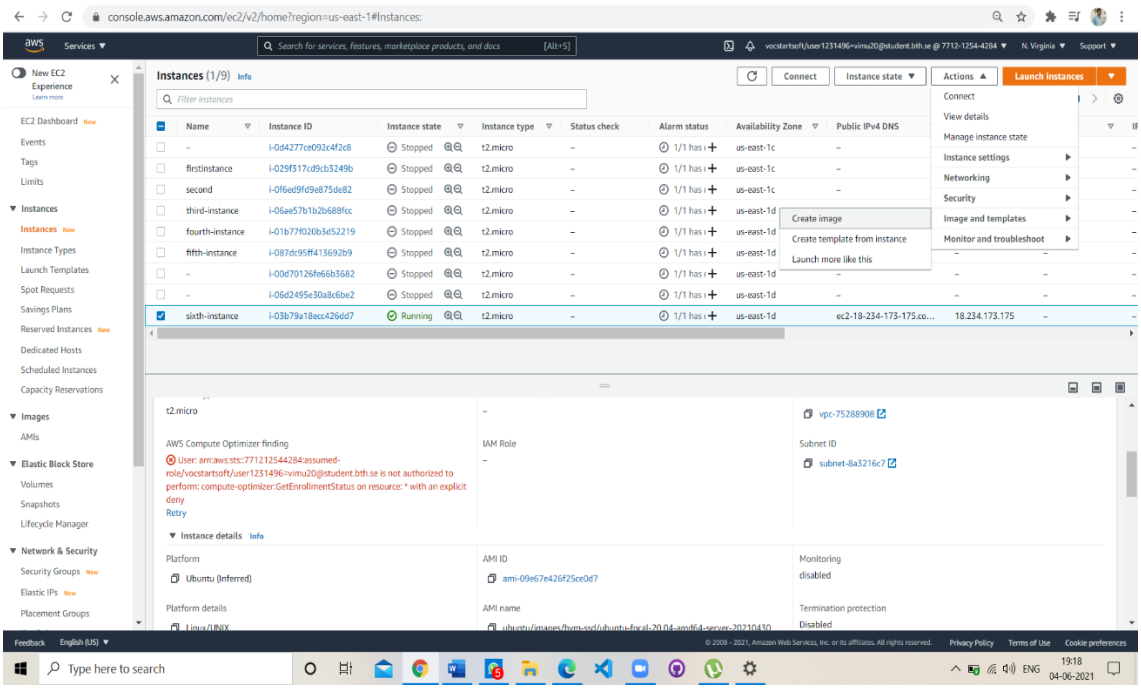
To test the flask app, the app made to run by giving the below command.

python3 app.py

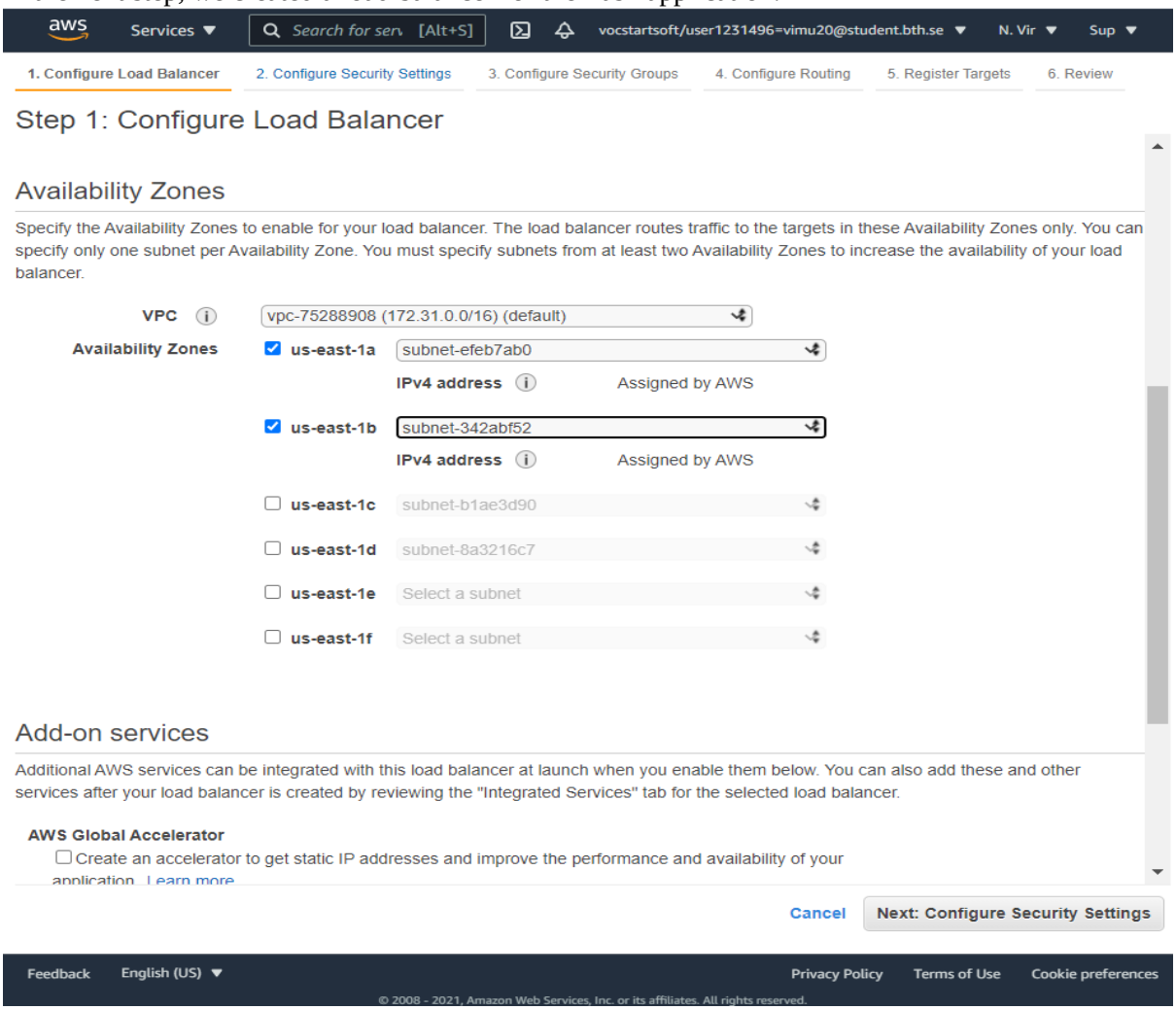
Using IPv4 address of the instance, the app runs on port 8080 in the browser.

Auto-Scaling flask app:

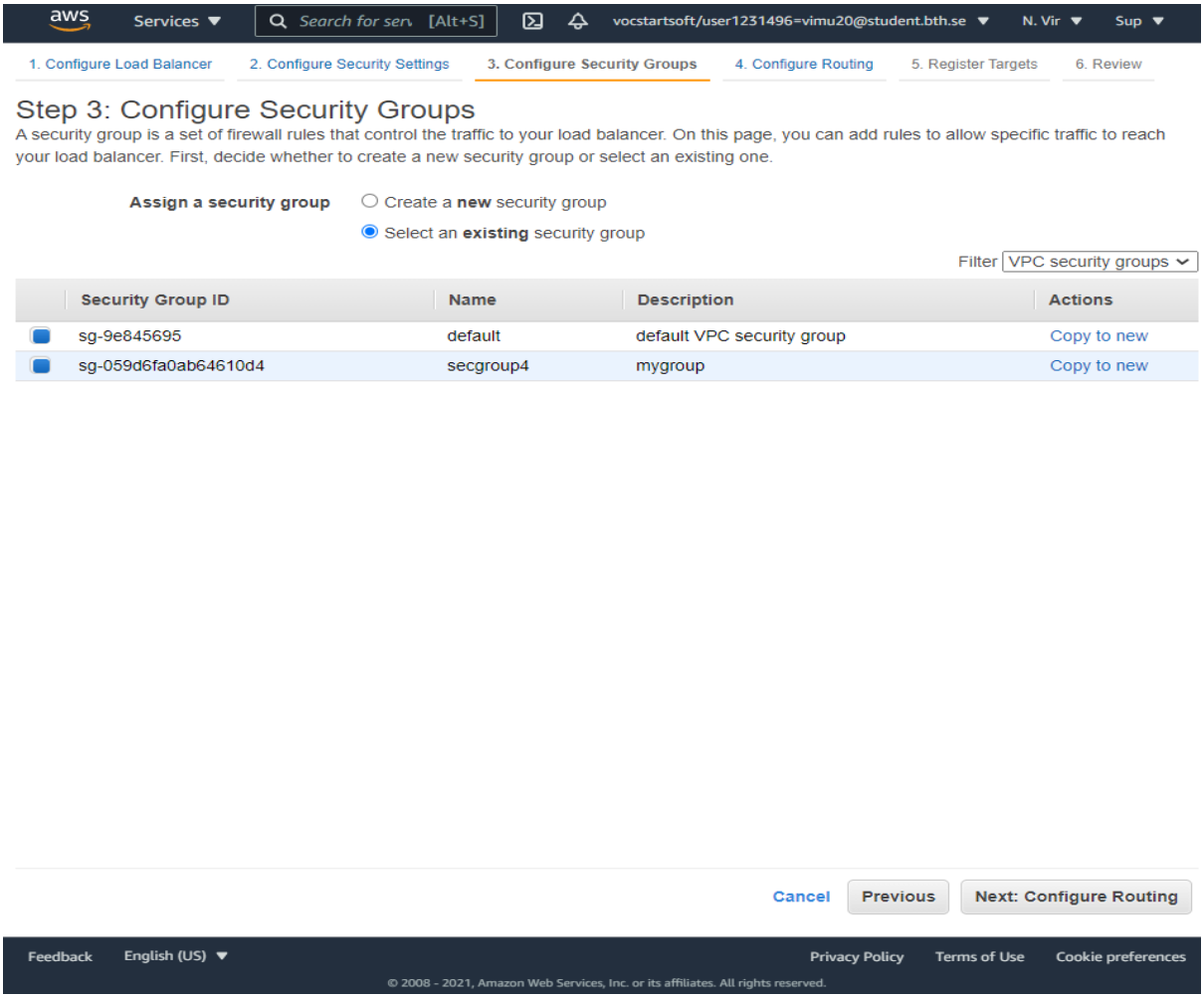
1. The first step is to create an image of the instance in which the flask application was deployed.



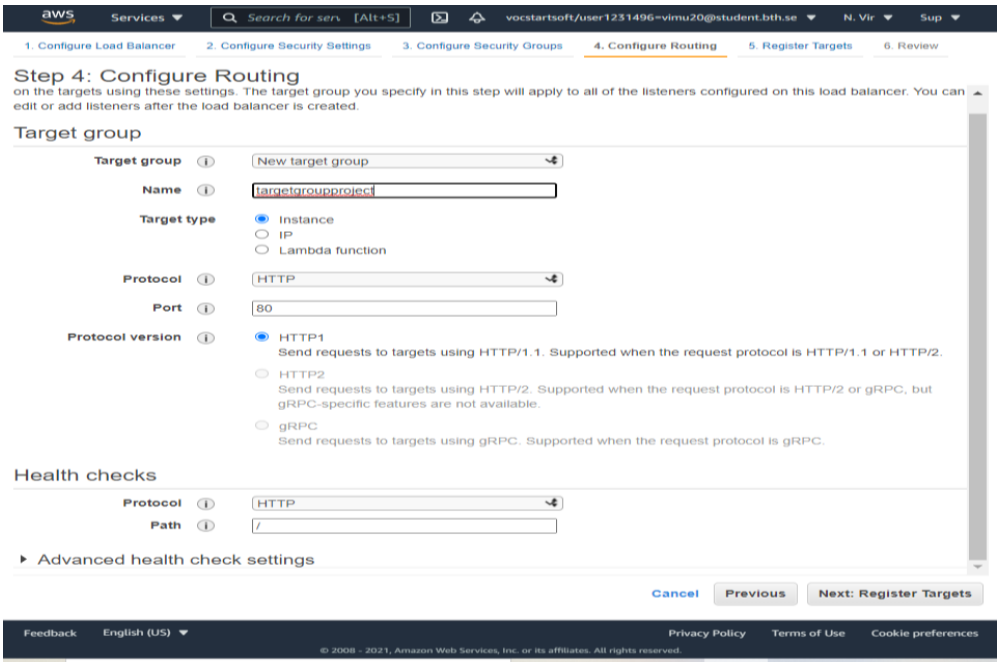
2. In the next step, we created a load balancer for the flask application.



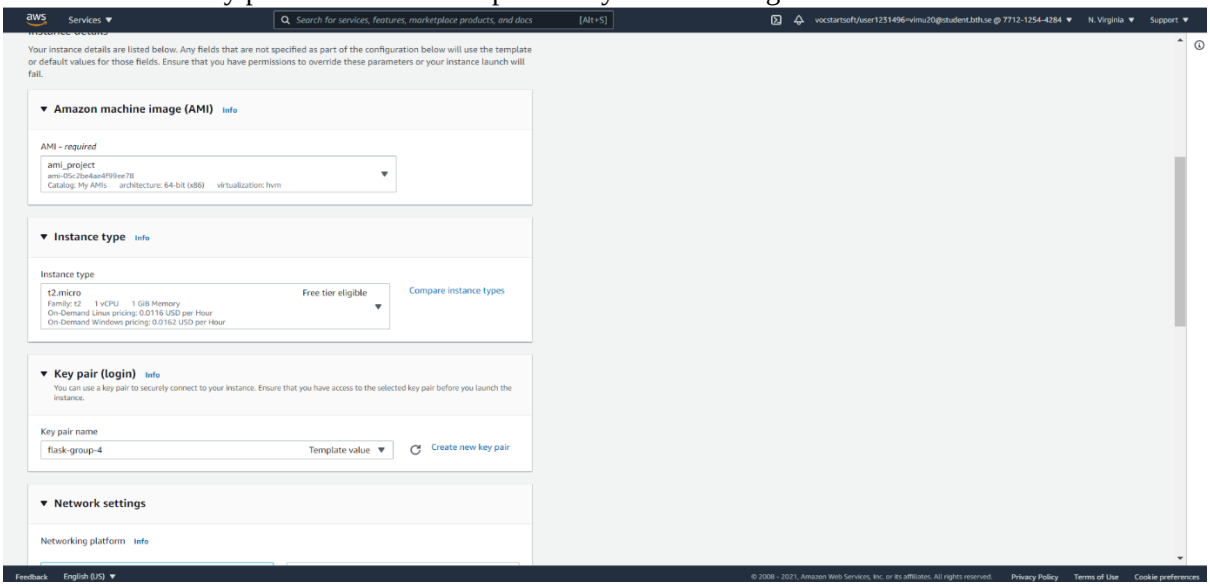
3. The security which was used before is configured and provided necessary inbound and outbound rules.

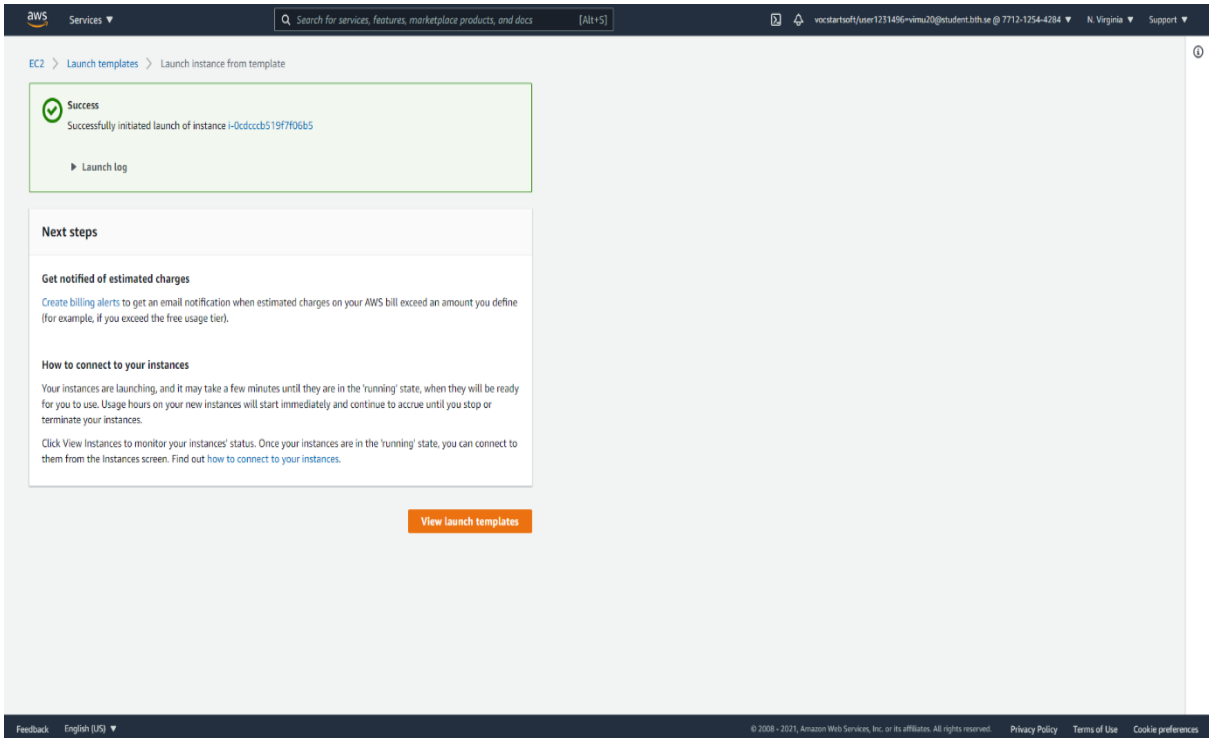


4. Then, a target group was created for the load balancer and then the load balancer for the application was launched.

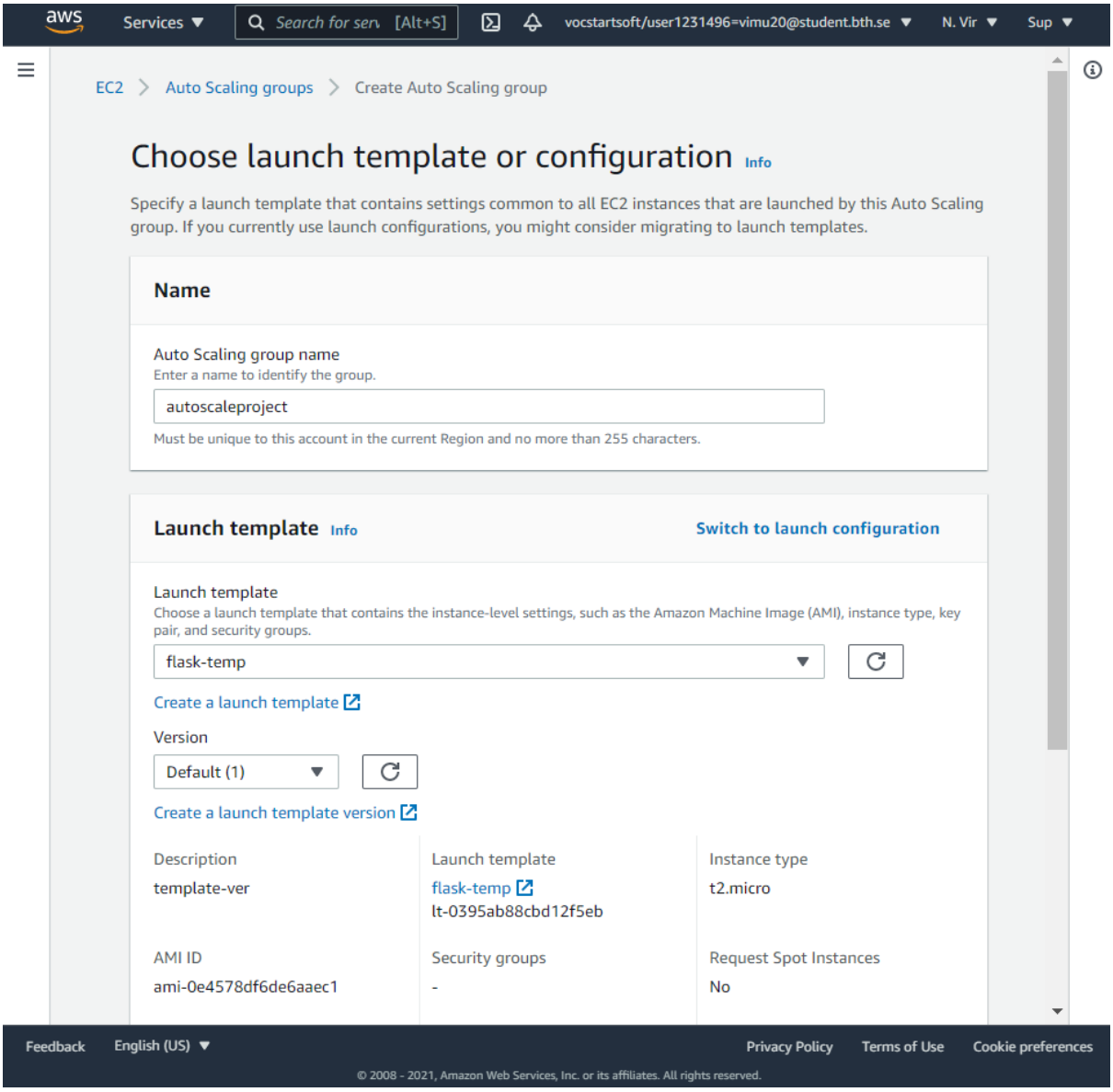


5. A launch configuration was created for the image which was created from the instance.
6. t2.micro instance type was selected since it is provided for free.
7. The existing security group was selected which was created and used before.
8. We selected the key pair which was used previously for launching instance.

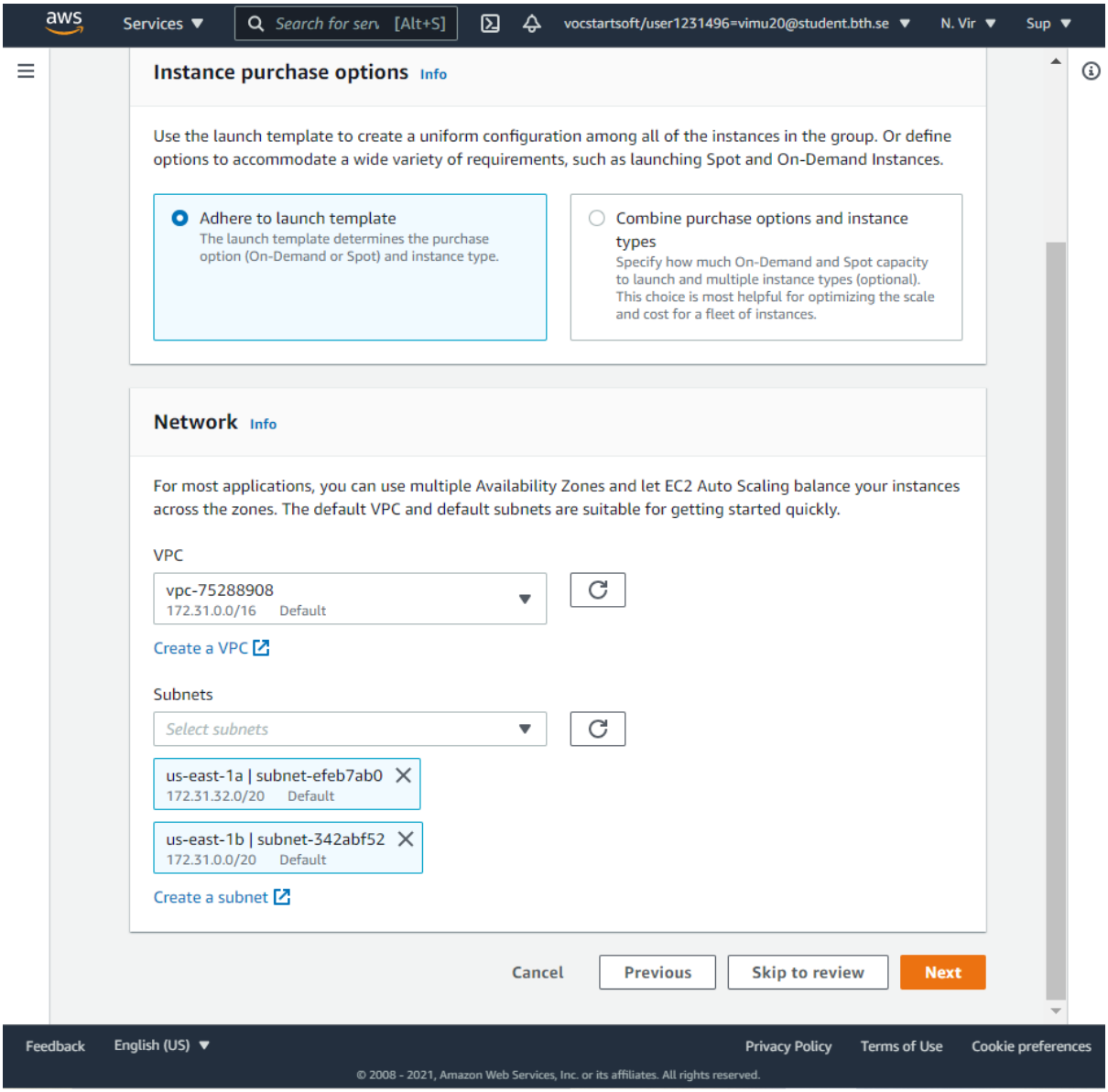




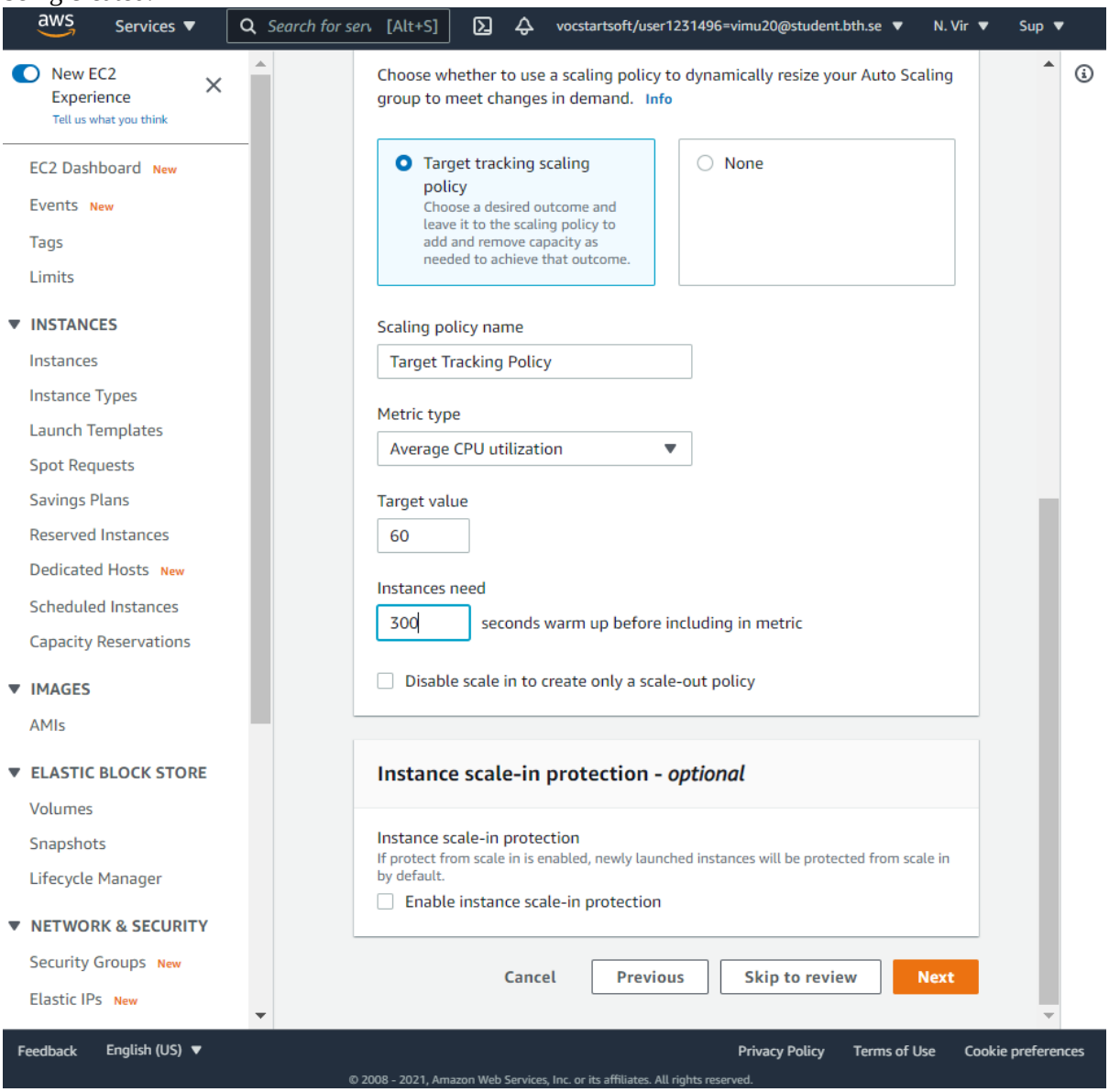
9. In the next step, we created an auto-scaling group by using the created launch configuration.

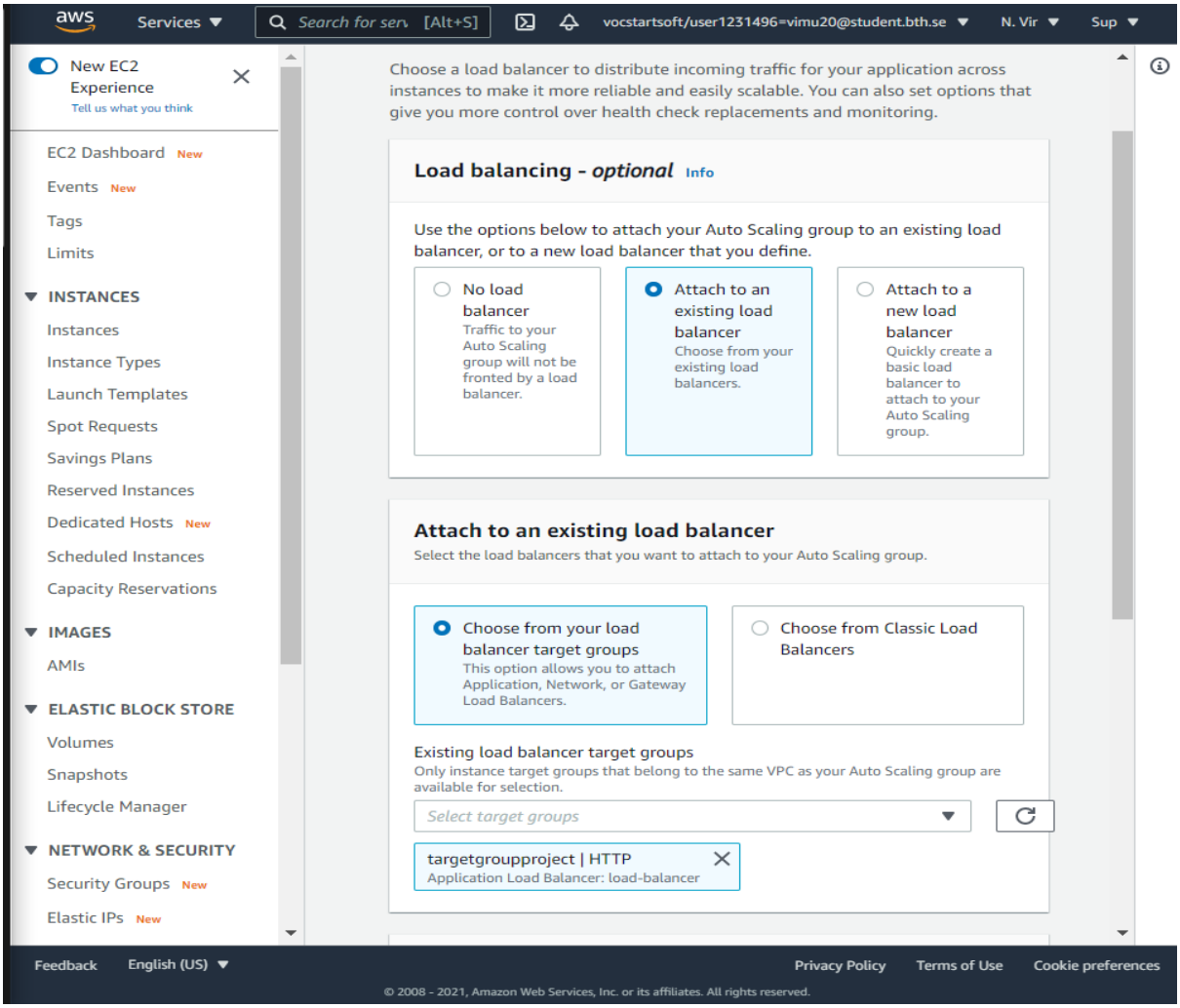


9. Since default vpc was selected, subnets were also selected as default provided.

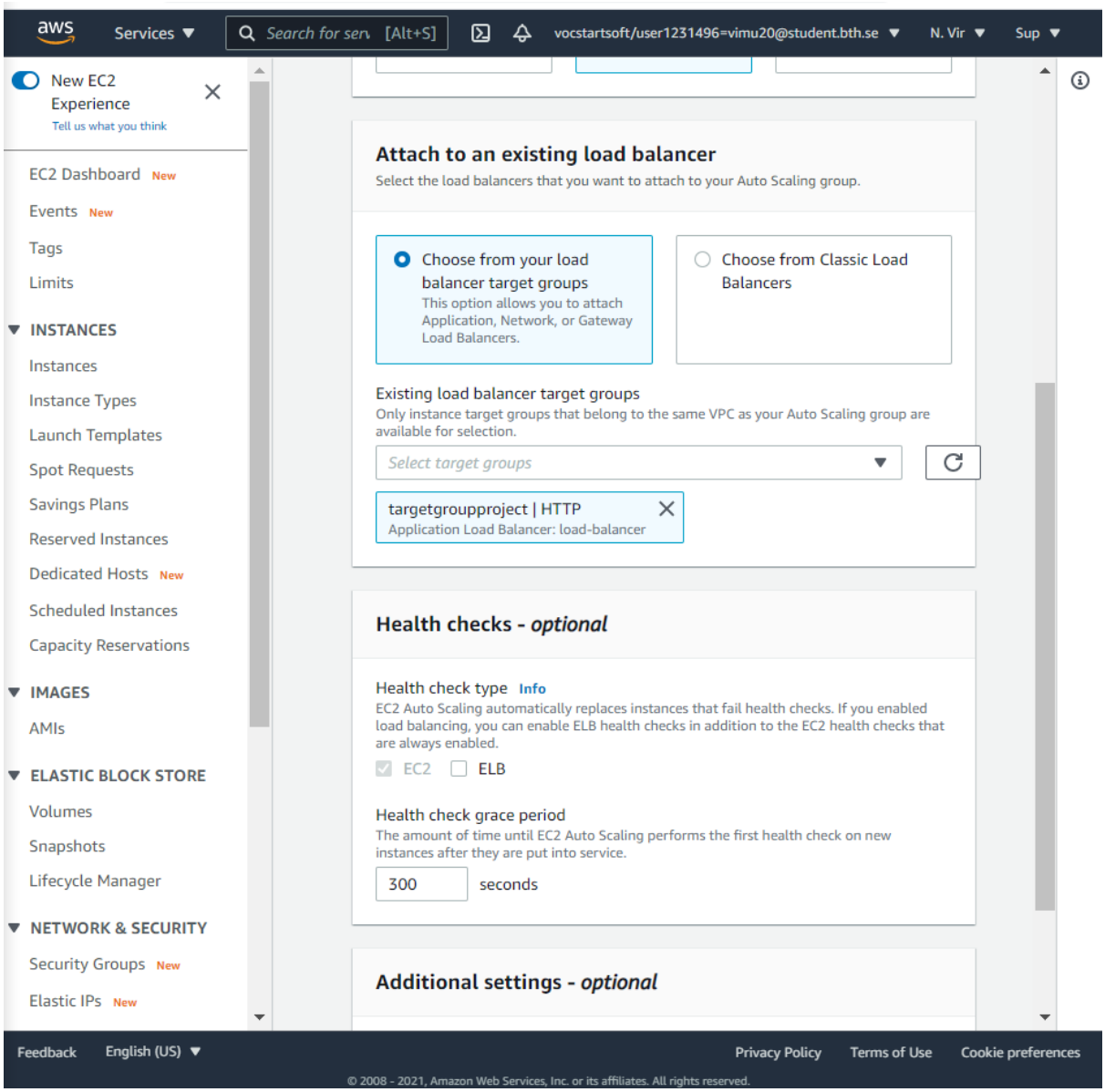

aws
Services
Search for ser [Alt+S]
vocstartsoft/user1231496=vimu20@student.bth.se
N. Vir
Sup
Instance purchase options Info
Use the launch template to create a uniform configuration among all of the instances in the group. Or define options to accommodate a wide variety of requirements, such as launching Spot and On-Demand Instances.
Adhere to launch template
The launch template determines the purchase option (On-Demand or Spot) and instance type.
Combine purchase options and instance types
Specify how much On-Demand and Spot capacity to launch and multiple instance types (optional). This choice is most helpful for optimizing the scale and cost for a fleet of instances.
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
vpc-75288908
172.31.0.0/16 Default
Create a VPC
Subnets
Select subnets
us-east-1a | subnet-efeb7ab0
172.31.32.0/20 Default
us-east-1b | subnet-342abf52
172.31.0.0/20 Default
Create a subnet
Cancel
Previous
Skip to review
Next
Feedback
English (US)
Privacy Policy
Terms of Use
Cookie preferences
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

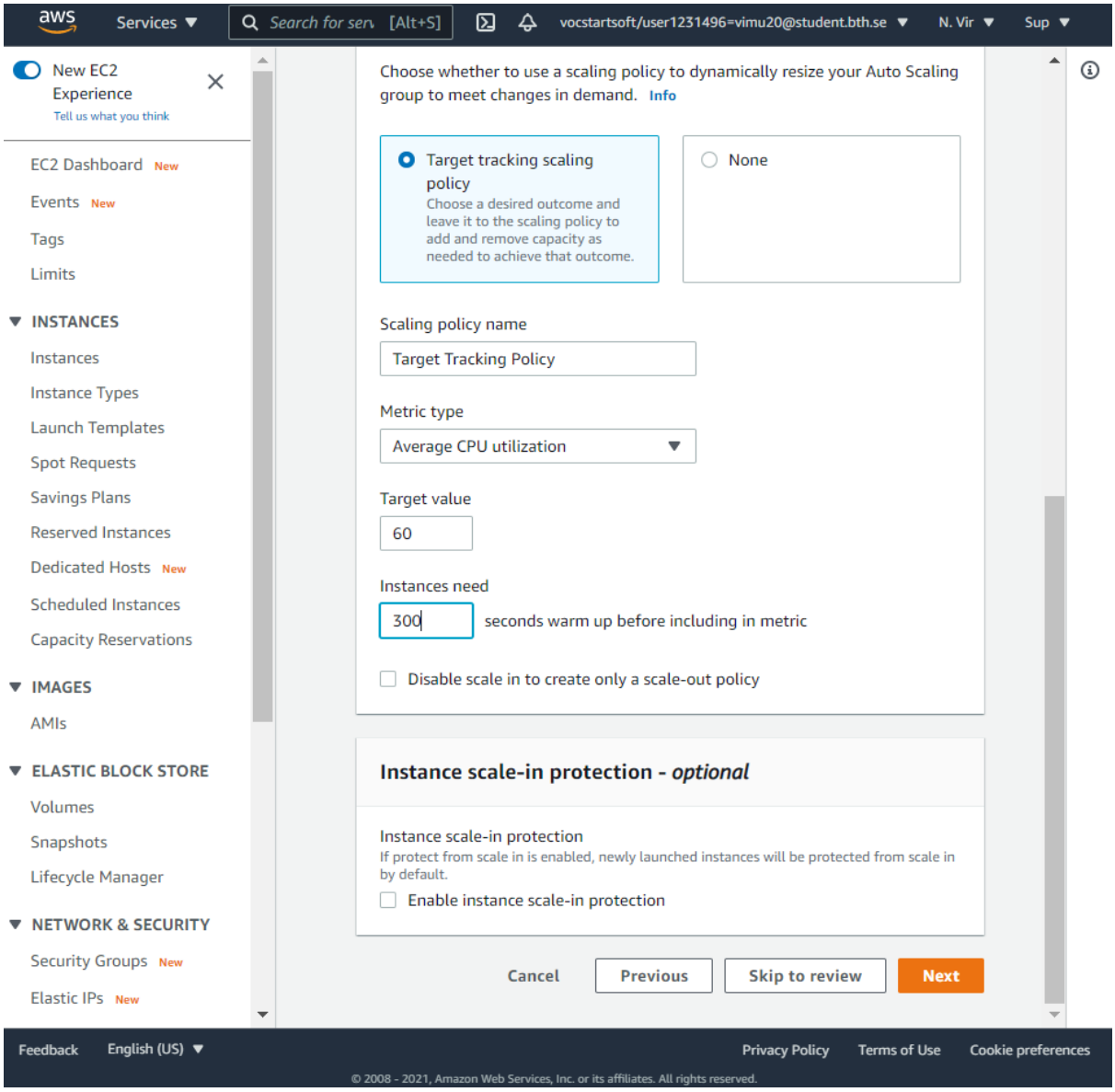
10. Then we attached the load balancer which was created before to the auto-scaling which is being created.



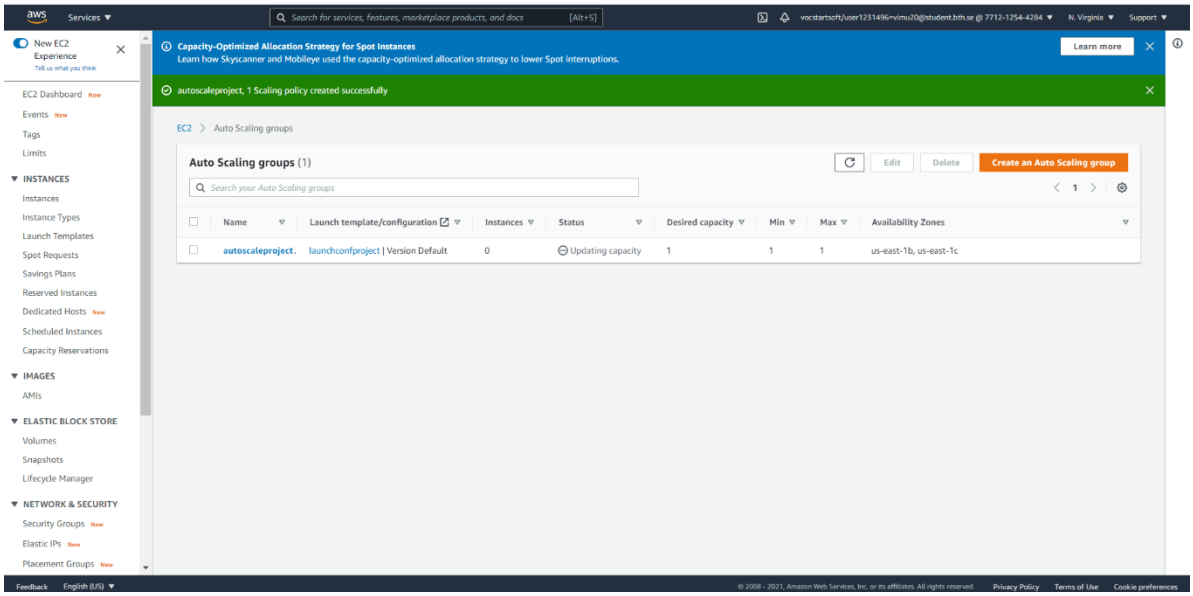


11. The group size is configured and scaling policies are added.
12. The desired capacity, minimum capacity and maximum capacity is selected by us.

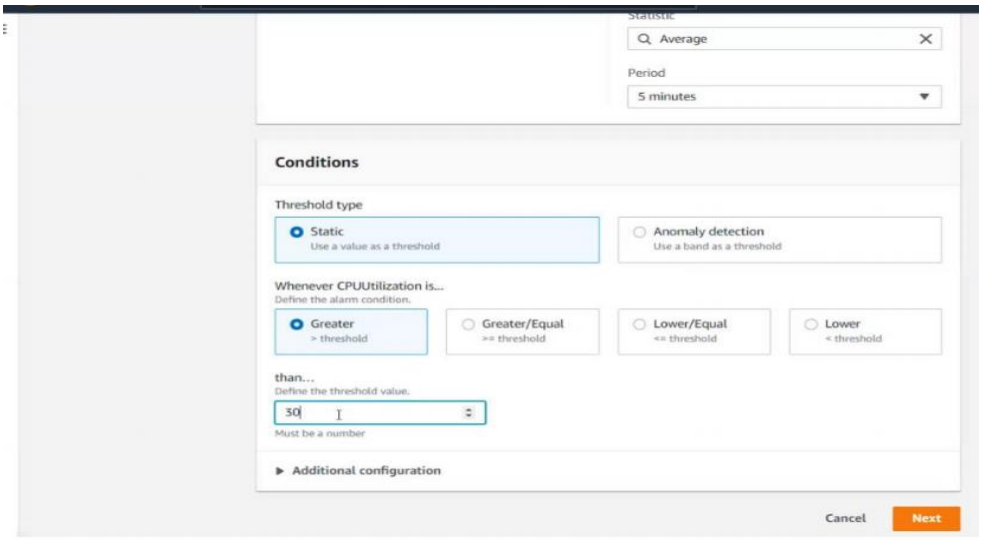

aws
Services
Search for ser [Alt+S]
vocstartsoft/user1231496=vimu20@student.bth.se
N. Vir
Sup
New EC2 Experience
Tell us what you think
EC2 Dashboard New
Events New
Tags
Limits
INSTANCES
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts New
Scheduled Instances
Capacity Reservations
IMAGES
AMIs
ELASTIC BLOCK STORE
Volumes
Snapshots
Lifecycle Manager
NETWORK & SECURITY
Security Groups New
Elastic IPs New
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.
Select target groups
targetgroupproject | HTTP
Application Load Balancer: load-balancer
Health checks - optional
Health check type Info
EC2 Auto Scaling automatically replaces instances that fail health checks. If you enabled load balancing, you can enable ELB health checks in addition to the EC2 health checks that are always enabled.
EC2
ELB
Health check grace period
The amount of time until EC2 Auto Scaling performs the first health check on new instances after they are put into service.
300
seconds
Additional settings - optional
Feedback
English (US)
Privacy Policy
Terms of Use
Cookie preferences
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.



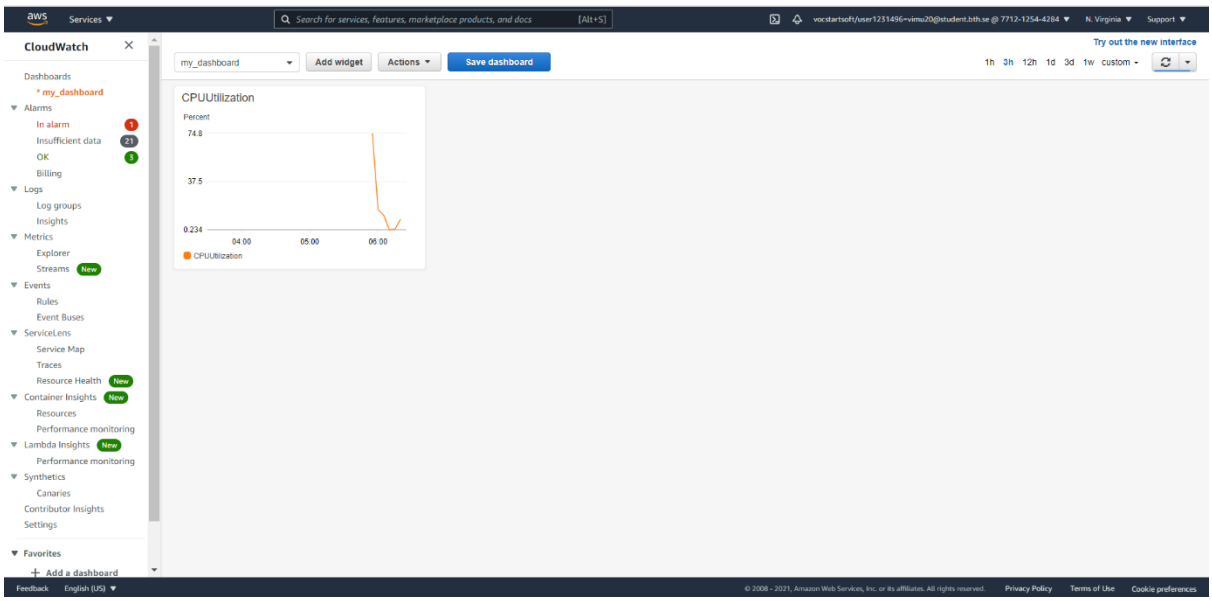
13. Finally Auto scaling group is created successfully as seen below.



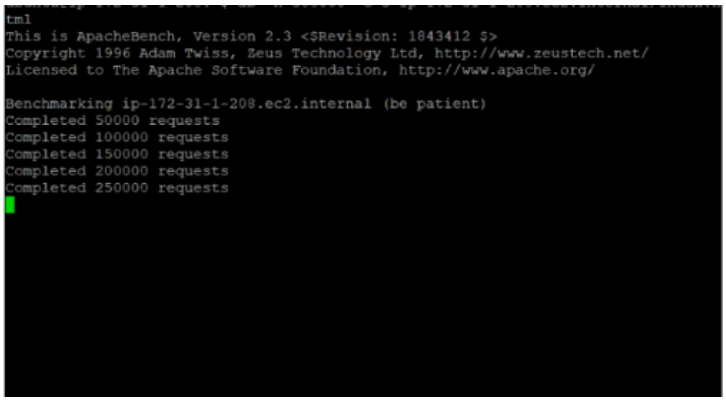
14. In the next step, we selected CPU utilization metric in the step scaling policy, and aa threshold value is set for the average CPU use, added a metric alarm and then added action and then created the step scaling policy.



15. In the next step, we created a dashboard in the amazon cloud watch and added metric CPU utilization for monitoring the instance.



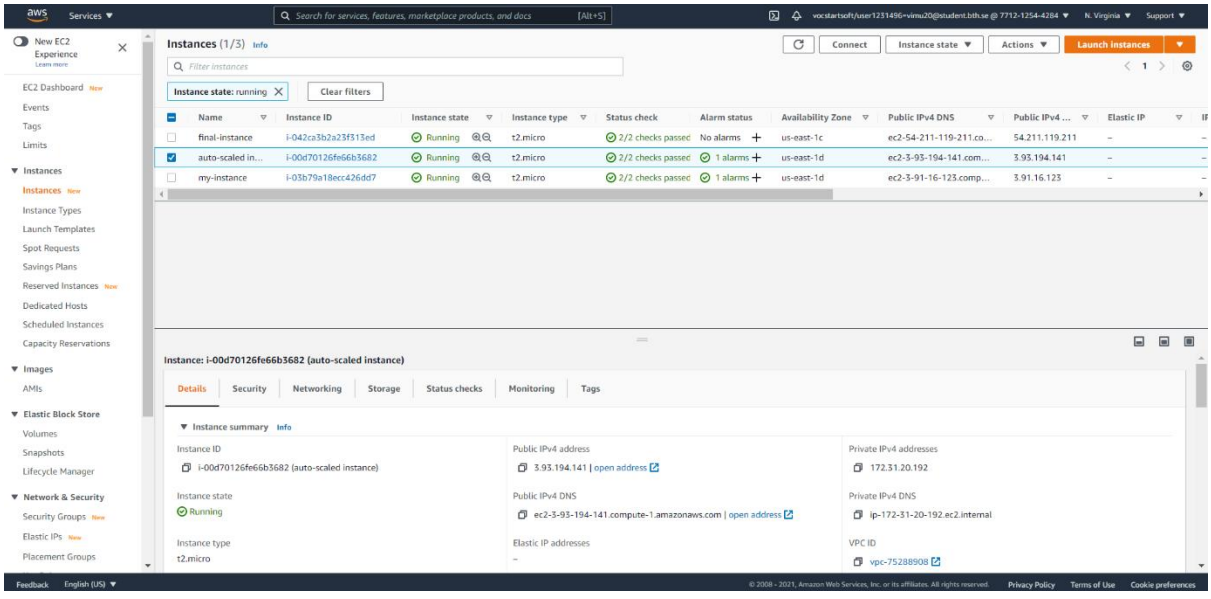
16. We stressed the CPU using apachebench. Below command increases CPU utilization
ab -n 500000 -c 5 ip-172-31-20-192.ec2.internal/index/html



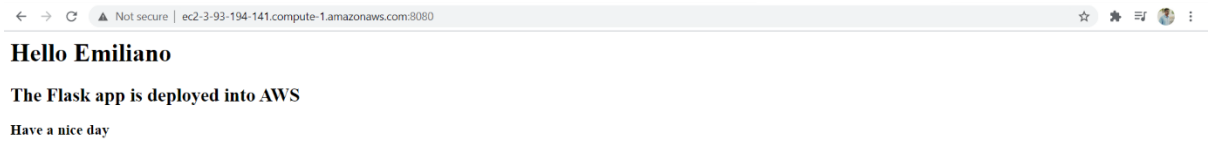
Validation

The static flask application is deployed into ubuntu server, and the application is accessible from Public IPv4 DNS address of the creates instance.

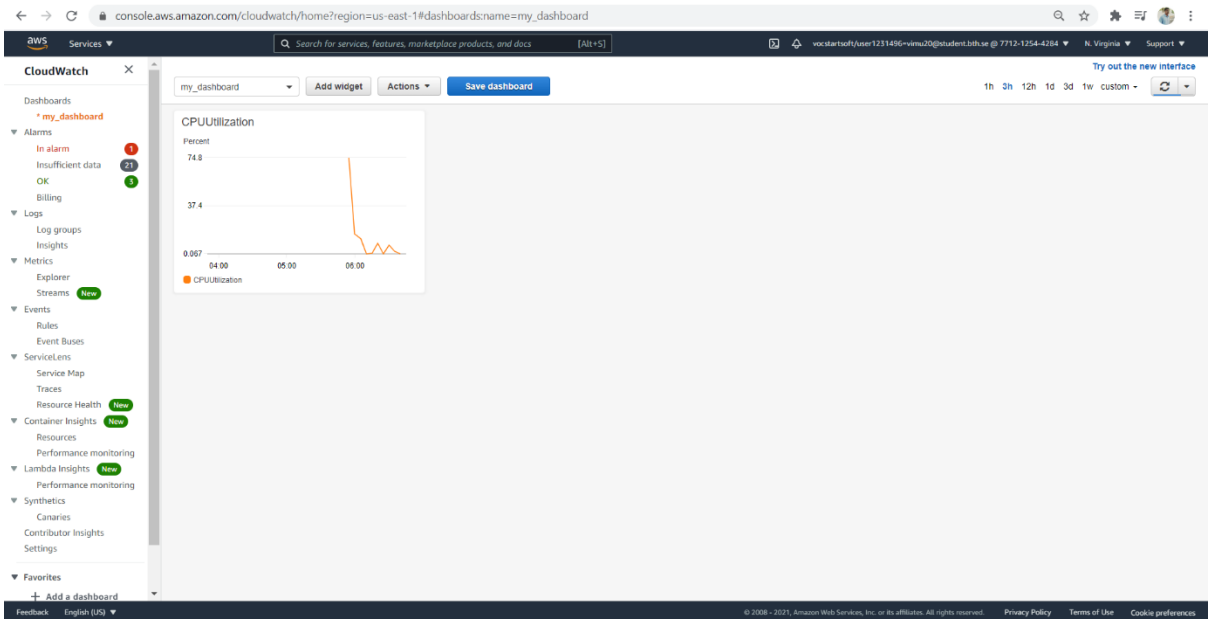
A desired capacity 2 is created in the auto-scaling group and launched with target tracking policy. This tracks the metric CPU utilization and then deployed this application using the AMI created from the instance. (ec2-3-93-194-141.compute-1.amazonaws.com:8080).



The app runs on the port 8080 which is a custom TCP rule set in the inbound rules of the security group.



The CPU utilization decreased due to increase in stress which is done using apachebench. This is monitored in dashboard created in amazon cloud watch service.



Results

1. The flask application was created in the ubuntu server itself instead of pulling it from github. Then the application runs in the AWS by configuring Nginx and Gunicorn web server. Using these improves the performance of the web app and makes the application to run in background.
2. The flask application deployed is auto scaled by adding necessary scaling policies with additional 2 instances that are configured according to the threshold value of the average CPU utilization of ubuntu server in which the flask application was deployed.
3. More stress decreases CPU utilization, triggering the alarm after reaching the threshold value. Additional instance was added in the dashboard and resulted in satisfying step scaling policy.

4. Hence, the flask application deployed in AWS fulfilled the scalability, and high availability with respect to computation.