

# 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.

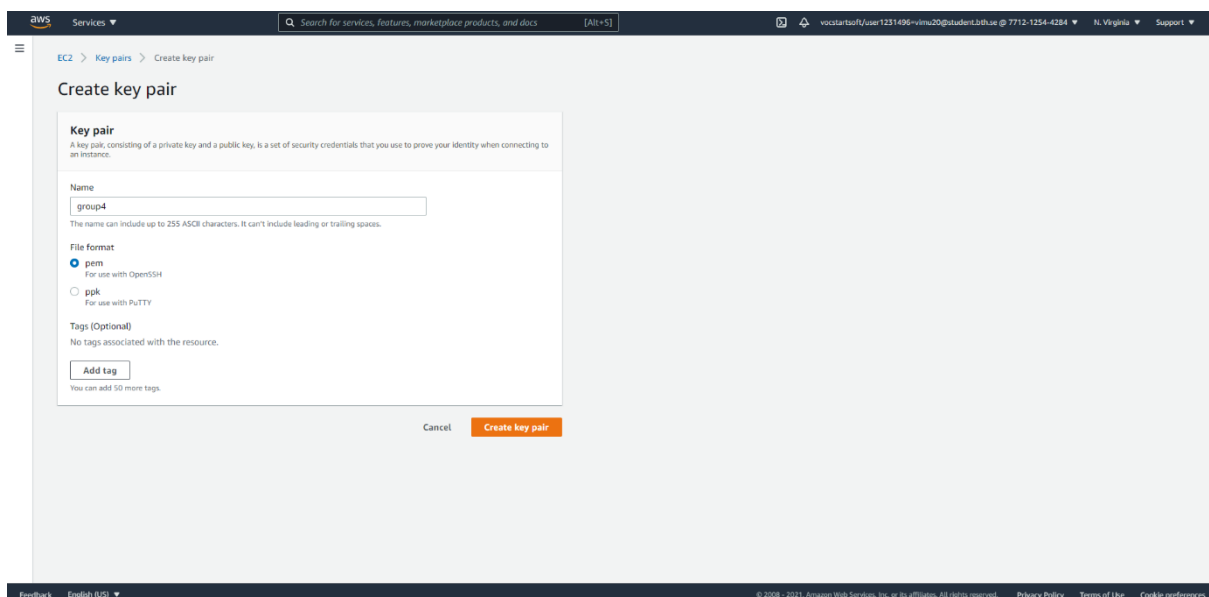
## Design Description

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.

**Create security group**

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

**Basic details**

Security group name:   
Name cannot be edited after creation.

Description:   
Allows SSH access to developers

VPC:

**Inbound rules**

This security group has no inbound rules.

**Outbound rules**

Type	Protocol	Port range	Destination	Description - optional
SSH	TCP	22	Anywhere	
HTTP	TCP	80	Anywhere	
Custom TCP	TCP	0	Custom	

**Tags - optional**

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

No tags associated with the resource.

**Create security group**

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.

awsServices

Search for services, features, marketplace products, and docs

[Alt+S]

vocstartof/user1231496~vims20@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?

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.

Hide

Try it out

SUSE Linux

Free tier eligible

SUSE Linux Enterprise Server 15 SP2 (HVM), SSD Volume Type - ami-0fde50fcbcd482077 (64-bit x86) / ami-05d25f76d89313bb (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.

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

Select

64-bit (x86)

64-bit (Arm)

Ubuntu

Free tier eligible

Ubuntu Server 20.04 LTS (HVM), SSD Volume Type - ami-09e67e428f25ce0d7 (64-bit x86) / ami-00d1ab6b335217cf (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).

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

Select

64-bit (x86)

64-bit (Arm)

Ubuntu

Free tier eligible

Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-0747bdcabd34c712a (64-bit x86) / ami-08353a25e80bee3e (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).

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

Select

64-bit (x86)

64-bit (Arm)

Windows

Free tier eligible

Microsoft Windows Server 2019 Base - ami-0fa0543b0171fe3

Microsoft Windows 2019 Datacenter edition. [English]

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

Select

64-bit (x86)

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

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

Select

64-bit (x86)

Feedback

English (US)

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

Privacy Policy

Terms of Use

Cookie preferences

awsServices

Search for services, features, marketplace products, and docs

[Alt+S]

vocstartof/user1231496~vims20@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 instances1Launch into Auto Scaling Group

Purchasing optionRequest Spot instances

Networkvpc-75288908 (default)Create new VPC

SubnetNo preference (default subnet in any Availability Zone)Create new subnet

Auto-assign Public IPUse subnet setting

Placement groupAdd instance to placement group

Capacity ReservationOpen

Domain join directoryNo directoryCreate new directory

IAM roleNoneCreate new IAM role

Shutdown behaviorStop

Stop - Hibernate behaviorEnable hibernation as an additional stop behavior

Enable termination protectionProtect against accidental termination

MonitoringEnable CloudWatch detailed monitoringAdditional charges apply

TenancyShared - Run on shared hardware instanceAdditional charges will apply for dedicated tenancy

Elastic InferenceAdd 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

awsServices

Search for services, features, marketplace products, and docs

[Alt+S]

vocstartof/user1231496~vims20@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	<input checked="" type="checkbox"/>	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

## 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
<input type="checkbox"/> sg-9e845695	default	default VPC security group	<a href="#">Copy to new</a>
<input checked="" type="checkbox"/> sg-02ec7d62473ea0c49	sec-group4	abdef	<a href="#">Copy to new</a>

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](#)

## 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](#)

**Instance Type** [Edit instance type](#)

**Security Groups** [Edit security groups](#)

**AMI Details**  
Ubuntu Server 20.04 LTS (HVM), SSD Volume Type - ami-09e67e426f25ce0d7  
Free tier eligible  
Root Device Type: ebs Virtualization type: hvm

**Instance Type**

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GiB)	EBS-Optimized Available	Network Performance
t2.micro	-	1	1	EBS only	-	Low to Moderate

**Security Groups**

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](#)

projct\_by\_group6.zip group4.pem [Show all](#)

Type here to search

aws Services Search for services, features, marketplace products, and docs [Alt+S] vocstartsoft/Asen1231496~vims20@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 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](#)

**Instance Type** [Edit instance type](#)

**Security Groups** [Edit security groups](#)

**AMI Details**  
Ubuntu Server 20.04 LTS (HVM), SSD Volume Type - ami-09e67e426f25ce0d7  
Free tier eligible  
Root Device Type: ebs Virtualization type: hvm

**Instance Type**

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GiB)	EBS-Optimized Available	Network Performance
t2.micro	-	1	1	EBS only	-	Low to Moderate

**Security Groups**

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](#)

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.

```
ubuntu@ip-172-31-20-192: ~/flaskapp
Microsoft Windows [Version 10.0.19041.985]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ADMIN>cd Desktop
C:\Users\ADMIN\Desktop>cd flask-aus
The system cannot find the path specified.

C:\Users\ADMIN\Desktop>ssh -i "group4.pem" root@ec2-54-211-119-211.compute-1.amazonaws.com
Warning: Identity file group4.pem not accessible: No such file or directory.
root@ec2-54-211-119-211.compute-1.amazonaws.com: Permission denied (publickey).

C:\Users\ADMIN\Desktop>cd flask-aus
C:\Users\ADMIN\Desktop\flask-aus>ssh -i "group4.pem" ubuntu@ec2-3-93-194-141.compute-1.amazonaws.com
The authenticity of host 'ec2-3-93-194-141.compute-1.amazonaws.com (3.93.194.141)' can't be established.
ECDSA key fingerprint is SHA256:9xzPhkCuj+eZKsgkd3y8K3bmFHaft58hIMdFVH/0.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-93-194-141.compute-1.amazonaws.com,3.93.194.141' (ECDSA) to the list of
known hosts.
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-1045-aws x86_64)

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

System information disabled due to load higher than 1.0

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-172-31-20-192:~$ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal InRelease
```

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 displays 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.

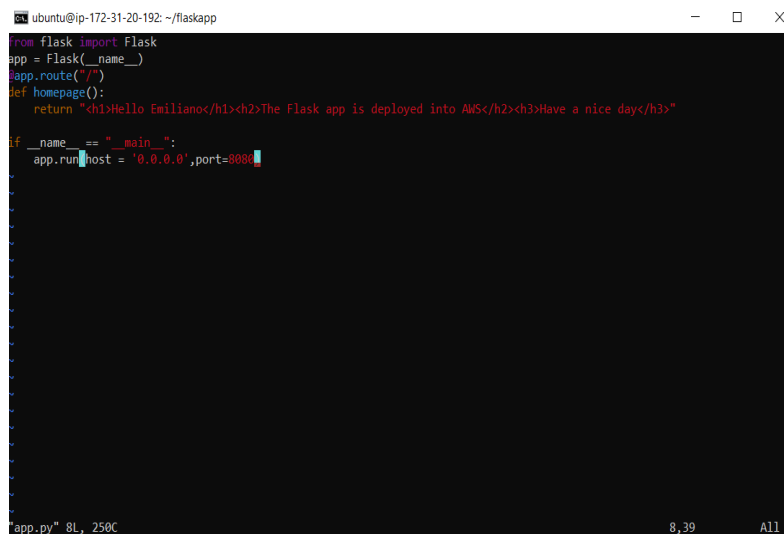
```
mkdir flaskapp
```

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
```



```
ubuntu@ip-172-31-20-192: ~/flaskapp
from flask import Flask
app = Flask(__name__)
app.route("/")
def homepage():
    return "<h1>Hello Emiliano</h1><h2>The Flask app is deployed into AWS</h2><h3>Have a nice day</h3>"
if __name__ == "__main__":
    app.run(host = '0.0.0.0',port=8080)
```

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.



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

The screenshot shows the AWS Management Console interface for configuring a load balancer. The top navigation bar includes the AWS logo, a search bar, and user information. The breadcrumb trail shows the steps: 1. Configure Load Balancer, 2. Configure Security Settings, 3. Configure Security Groups (current step), 4. Configure Routing, 5. Register Targets, and 6. Review.

### Step 3: Configure Security Groups

A security group is a set of firewall rules that control the traffic to your load balancer. On this page, you can add rules to allow specific traffic to reach your load balancer. First, decide whether to create a new security group or select an existing one.

**Assign a security group**

- ☐ Create a **new** security group
- ☒ Select an **existing** security group

Filter: VPC security groups

Security Group ID	Name	Description	Actions
<input checked="" type="checkbox"/> sg-9e845695	default	default VPC security group	<a href="#">Copy to new</a>
<input checked="" type="checkbox"/> sg-059d6fa0ab64610d4	secgroup4	mygroup	<a href="#">Copy to new</a>

Navigation buttons: [Cancel](#), [Previous](#), [Next: Configure Routing](#)

Footer: Feedback, English (US), Privacy Policy, Terms of Use, Cookie preferences, © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

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

The screenshot shows the AWS Management Console interface for configuring routing. The top navigation bar is consistent with the previous screenshot. The breadcrumb trail shows the steps: 1. Configure Load Balancer, 2. Configure Security Settings, 3. Configure Security Groups, 4. Configure Routing (current step), 5. Register Targets, and 6. Review.

### Step 4: Configure Routing

on the targets using these settings. The target group you specify in this step will apply to all of the listeners configured on this load balancer. You can edit or add listeners after the load balancer is created.

**Target group**

Target group:

Name:

Target type:

- ☒ Instance
- ☐ IP
- ☐ Lambda function

Protocol:

Port:

Protocol version:

- ☒ HTTP1  
Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.
- ☐ HTTP2  
Send requests to targets using HTTP/2. Supported when the request protocol is HTTP/2 or gRPC, but gRPC-specific features are not available.
- ☐ gRPC  
Send requests to targets using gRPC. Supported when the request protocol is gRPC.

**Health checks**

Protocol:

Path:

Advanced health check settings

Navigation buttons: [Cancel](#), [Previous](#), [Next: Register Targets](#)

Footer: Feedback, English (US), Privacy Policy, Terms of Use, Cookie preferences, © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.



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.

Amazon machine image (AMI) info

AMI - required

ami\_project  
ami-05c2b6aa4f99e78  
Catalog: My AMIs architecture: 64-bit (x86) virtualization: hvm

Instance type info

Instance type

t2.micro Free tier eligible Compare instance types  
Family: t2 1 vCPU 1 GB Memory  
On-Demand Linux pricing: 0.0116 USD per Hour  
On-Demand Windows pricing: 0.0162 USD per Hour

Key pair (login) info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name  
flask-group-4 Template value Create new key pair

Network settings

Networking platform info

EC2 > Launch templates > Launch instance from template

Success

Successfully initiated launch of instance i-0cdcccb519f7d06b5

Launch log

Next steps

Get notified of estimated charges

Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).

How to connect to your instances

Your instances are launching, and it may take a few minutes until they are in the 'running' state, when they will be ready for you to use. Usage hours on your new instances will start immediately and continue to accrue until you stop or terminate your instances.

Click View instances to monitor your instances' status. Once your instances are in the 'running' state, you can connect to them from the Instances screen. Find out how to connect to your instances.

View launch templates

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

aws

Services

Search for serv. [Alt+S]

vocstartsoft/user1231496=vimu20@student.bth.se

N. Vir

Sup

EC2 > Auto Scaling groups > Create Auto Scaling group

## Choose launch template or configuration [Info](#)

Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group. If you currently use launch configurations, you might consider migrating to launch templates.

### Name

Auto Scaling group name

Enter a name to identify the group.

autoscaleproject

Must be unique to this account in the current Region and no more than 255 characters.

### Launch template [Info](#)

[Switch to launch configuration](#)

Launch template

Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.

flask-temp

[Create a launch template](#)

Version

Default (1)

[Create a launch template version](#)

Description	Launch template	Instance type
template-ver	<a href="#">flask-temp</a> lt-0395ab88cbd12f5eb	t2.micro
AMI ID	Security groups	Request Spot Instances
ami-0e4578df6de6aacc1	-	No

Feedback

English (US)

Privacy Policy

Terms of Use

Cookie preferences

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

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

aws

Services

Search for serv [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.

The screenshot shows the AWS Management Console interface for creating a new Auto Scaling group. The left sidebar contains navigation links for EC2 Dashboard, Events, Tags, Limits, INSTANCES, IMAGES, ELASTIC BLOCK STORE, and NETWORK & SECURITY. The main content area displays the 'Create New Auto Scaling Group' wizard. The 'Target tracking scaling policy' is selected, and the scaling policy name is 'Target Tracking Policy'. The metric type is 'Average CPU utilization', the target value is '60', and the instances need '300' seconds warm up. The 'Instance scale-in protection' section is optional and currently disabled. The bottom of the console shows the 'Cancel', 'Previous', 'Skip to review', and 'Next' buttons.

Choose whether to use a scaling policy to dynamically resize your Auto Scaling group to meet changes in demand. [Info](#)

☒ Target tracking scaling policy  
Choose a desired outcome and leave it to the scaling policy to add and remove capacity as needed to achieve that outcome.

☐ None

Scaling policy name  
Target Tracking Policy

Metric type  
Average CPU utilization

Target value  
60

Instances need  
300 seconds warm up before including in metric

☐ Disable scale in to create only a scale-out policy

**Instance scale-in protection - optional**

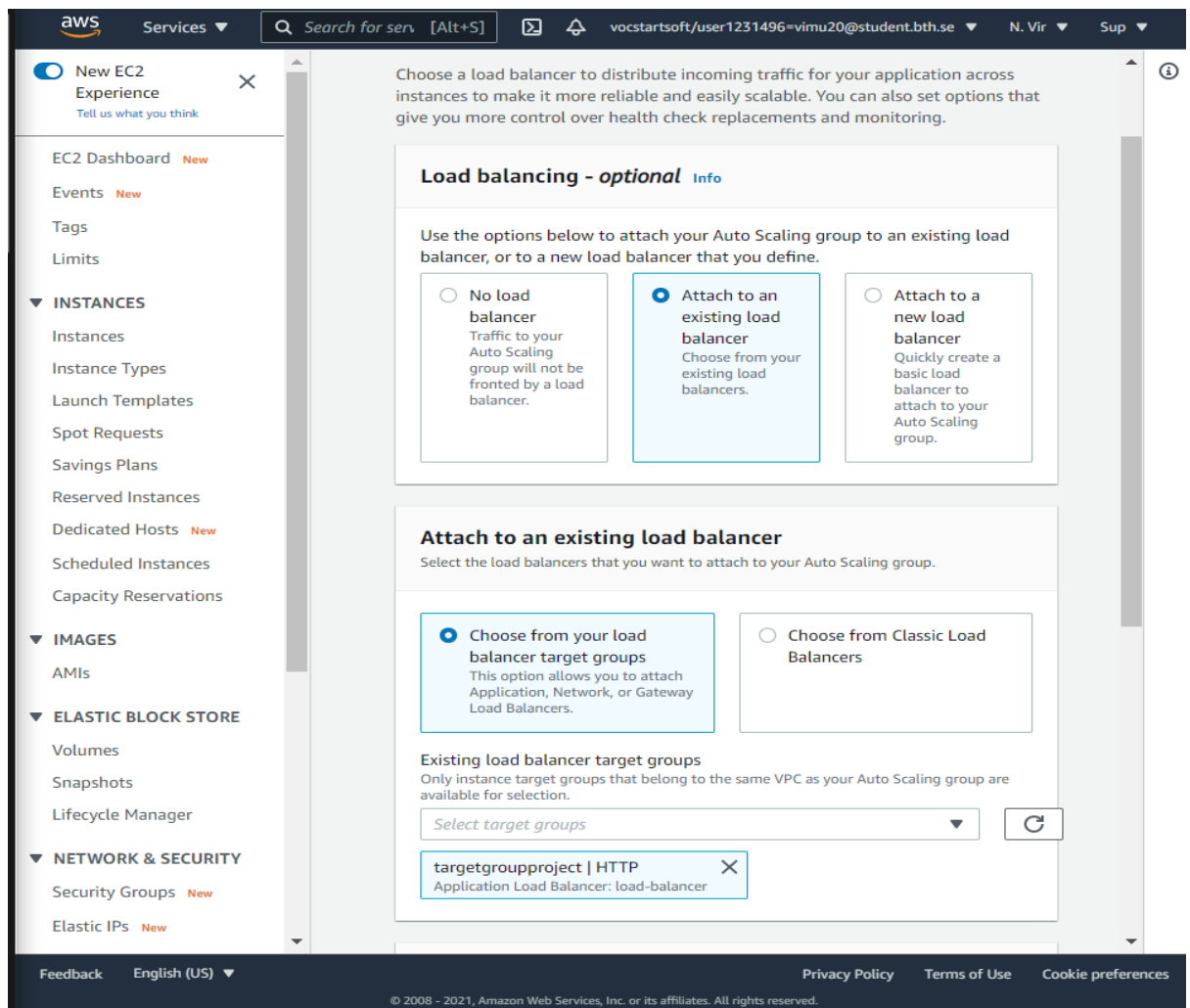
Instance scale-in protection  
If protect from scale in is enabled, newly launched instances will be protected from scale in by default.

☐ Enable instance scale-in protection

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.



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 serv [Alt+S]

vocstartsoft/user1231496=vimu20@student.bth.se

N. Vir

Sup

New EC2 Experience

EC2 Dashboard

Events

Tags

Limits

INSTANCES

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Scheduled Instances

Capacity Reservations

IMAGES

AMIs

ELASTIC BLOCK STORE

Volumes

Snapshots

Lifecycle Manager

NETWORK & SECURITY

Security Groups

Elastic IPs

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

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.

**aws** Services  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**

Choose whether to use a scaling policy to dynamically resize your Auto Scaling group to meet changes in demand. [Info](#)

☒ **Target tracking scaling policy**  
Choose a desired outcome and leave it to the scaling policy to add and remove capacity as needed to achieve that outcome.

☐ None

Scaling policy name

Metric type

Target value

Instances need  
 seconds warm up before including in metric

☐ Disable scale in to create only a scale-out policy

**Instance scale-in protection - optional**

Instance scale-in protection  
If protect from scale in is enabled, newly launched instances will be protected from scale in by default.

☐ Enable instance scale-in protection

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.

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

**aws** Services  vocstartsoft/user1231496=vimu20@student.bth.se 7712-1254-4234 N. Virginia Support

**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**

Placement Groups **New**

**Capacity-Optimized Allocation Strategy for Spot Instances**  
Learn how SkyScanner and Mobleye used the capacity-optimized allocation strategy to lower Spot interruptions. [Learn more](#)

**autoscaleproject, 1 Scaling policy created successfully**

EC2 > Auto Scaling groups

**Auto Scaling groups (1)** Refresh Edit Delete Create an Auto Scaling group

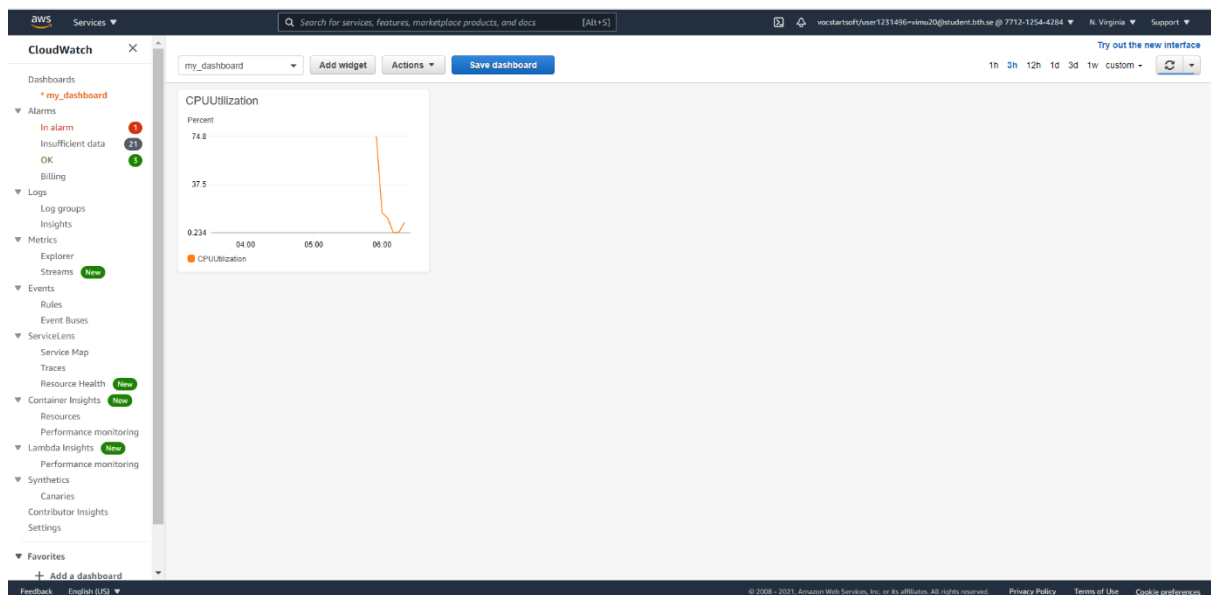
<input type="checkbox"/>	Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zones
<input type="checkbox"/>	autoscaleproject	launchconfproject   Version Default	0	Updating capacity	1	1	1	us-east-1b, us-east-1c

Feedback English (US) © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

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

The screenshot shows the 'Conditions' section of the AWS CloudWatch console. At the top, the 'Statistic' is set to 'Average' and the 'Period' is '5 minutes'. Under 'Threshold type', 'Static' is selected with the subtext 'Use a value as a threshold'. Below this, the condition is defined as 'Whenever CPUUtilization is... Greater > threshold'. The 'than...' field contains the value '50', with a note 'Must be a number'. At the bottom right are 'Cancel' and 'Next' buttons.

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

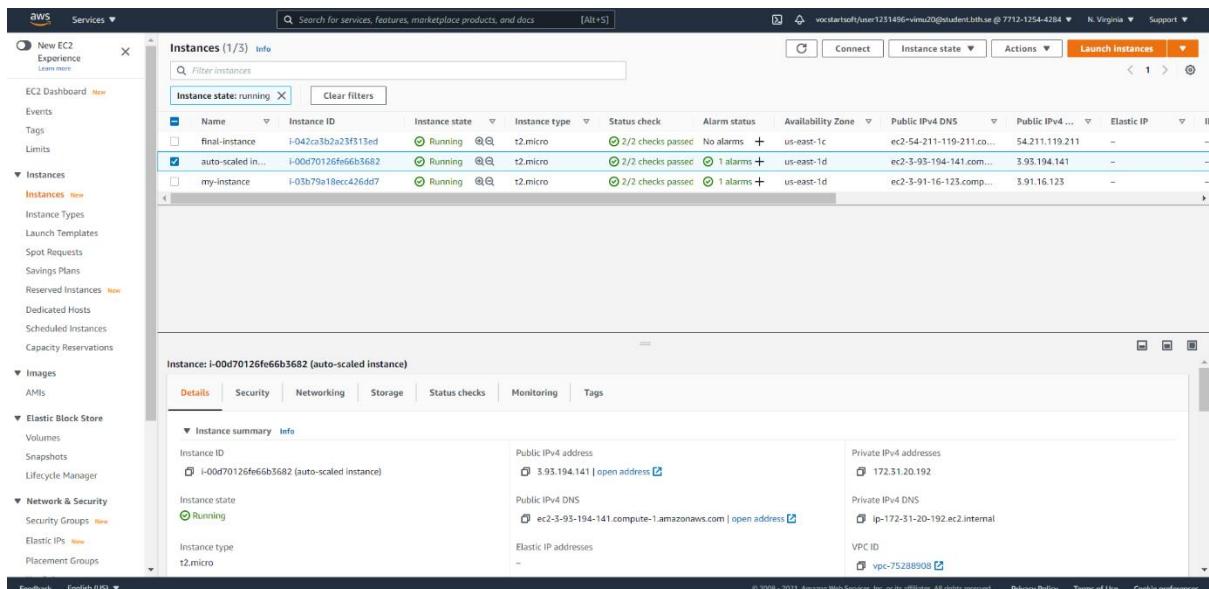
```
tml
This is ApacheBench, Version 2.3 <Revision: 1843412>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking ip-172-31-1-208.ec2.internal (be patient)
Completed 50000 requests
Completed 100000 requests
Completed 150000 requests
Completed 200000 requests
Completed 250000 requests
█
```

## 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.

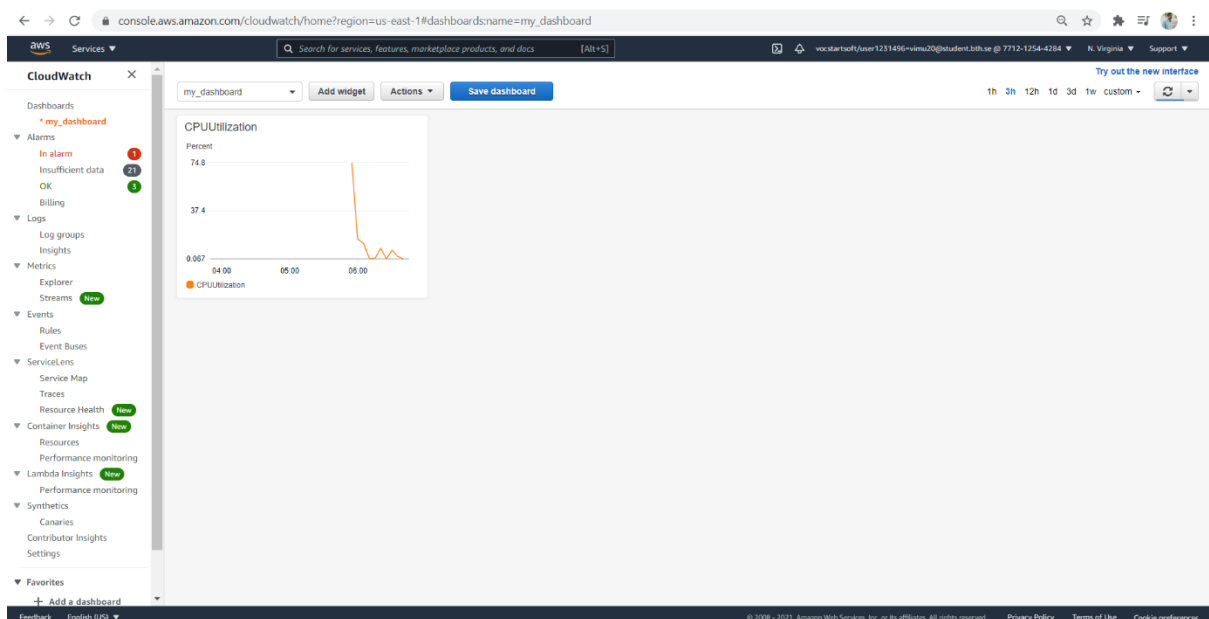
← → ↻ 🔒 Not secure | ec2-3-93-194-141.compute-1.amazonaws.com:8080 ☆ ⚙️ 📄 👤 ⋮

# Hello Emiliano

The Flask app is deployed into AWS

Have a nice day

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.