

# DV1566 Cloud Computing

## Project Report

### Group 9

Name	Personal Number
Devi Priya Battina	19990319-T121
Simron Padhi	19980821-T321
Surya Teja Kodukulla	19981010-T130

## Problem Description

This project aims to implement a Web application on the AWS (Amazon Web Services) platform where the application must be able to handle many users accessing the data in the application by means of auto scaling and satisfy the scalability requirements.

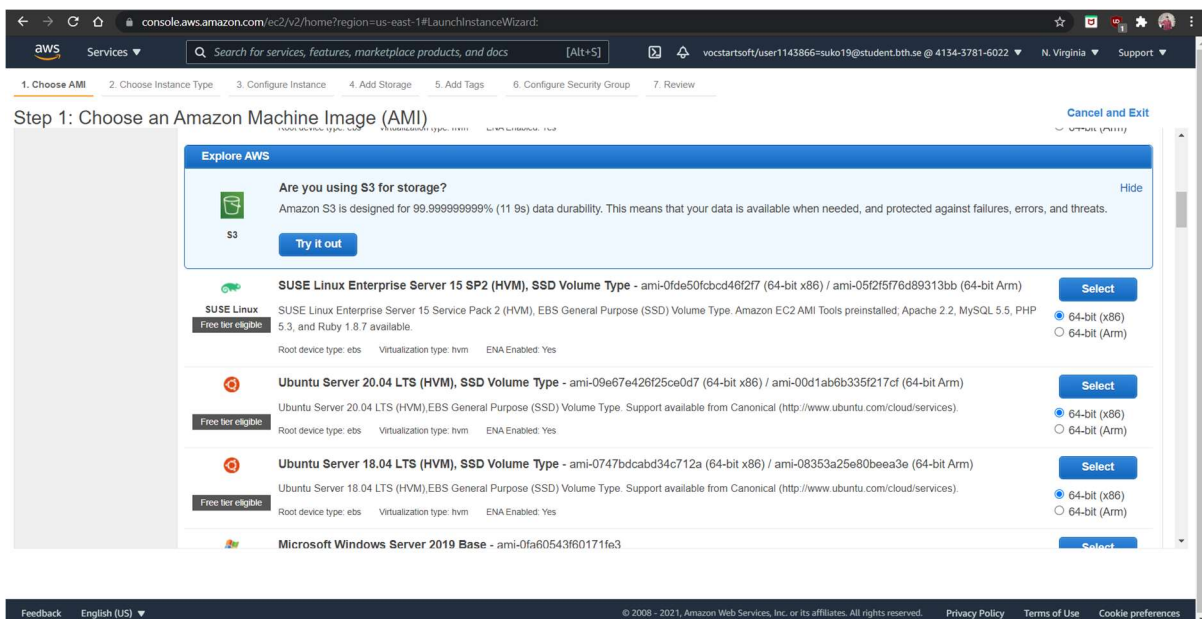
## Design Method

A simple web page is created and deployed into AWS platform. The implementation is achieved using the EC2 instance of the Ubuntu Amazon Machine Image in the server. Along with Ubuntu, Apache Web Server and Gunicorn WSGI are configured to make sure that the application runs even if the instance closes and used to deploy the application. Scaling policies are added in order to handle the requests for the data in the application and auto scaling is achieved using the scaling policies.

## Implementation

Step 1: An EC2 instance is launched in the Amazon Web Services to deploy the web application created.

- a. Select the EC2 services and launch it in AWS console
- b. The Ubuntu Amazon Machine Image is selected.



- c. The default VPC and subnets are selected.
- d. A security group is configured with adequate rules.
- e. A key pair is generated with desired name and the PEM file stored.

← → ↻ 🏠 console.aws.amazon.com/ec2/v2/home?region=us-east-1#LaunchInstanceWizard: 🔍 Search for services, features, marketplace products, and docs [Alt+S] 📄 📧 vocstartsoft/user1143866=suko19@student.bth.se @ 4134-3781-6022 📍 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 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Filter by: All instance families Current generation Show/Hide Columns

Currently selected: t2.micro (- ECUs, 1 vCPUs, 2.5 GHz, -, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GiB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input type="checkbox"/>	t2	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	t2	t2.micro <small>Free tier eligible</small>	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.large	2	8	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.xlarge	4	16	EBS only	-	Moderate	Yes
<input type="checkbox"/>	t2	t2.2xlarge	8	32	EBS only	-	Moderate	Yes
<input type="checkbox"/>	t3	t3.nano	2	0.5	EBS only	Yes	Up to 5 Gigabit	Yes

Cancel Previous **Review and Launch** Next: Configure Instance Details

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

← → ↻ 🏠 console.aws.amazon.com/ec2/v2/home?region=us-east-1#LaunchInstanceWizard: 🔍 Search for services, features, marketplace products, and docs [Alt+S] 📄 📧 vocstartsoft/user1143866=suko19@student.bth.se @ 4134-3781-6022 📍 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-f4a77989 (default) Create new VPC

Subnet No preference (default subnet in any Availability Zone) Create new subnet

Auto-assign Public IP Use subnet setting (Enable)

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

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

← → ↻ 🏠 console.aws.amazon.com/ec2/v2/home?region=us-east-1#LaunchInstanceWizard: 🔍 Search for services, features, marketplace products, and docs [Alt+S] 📄 📧 vocstartsoft/user1143866=suko19@student.bth.se @ 4134-3781-6022 📍 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 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 name: launch-wizard-1

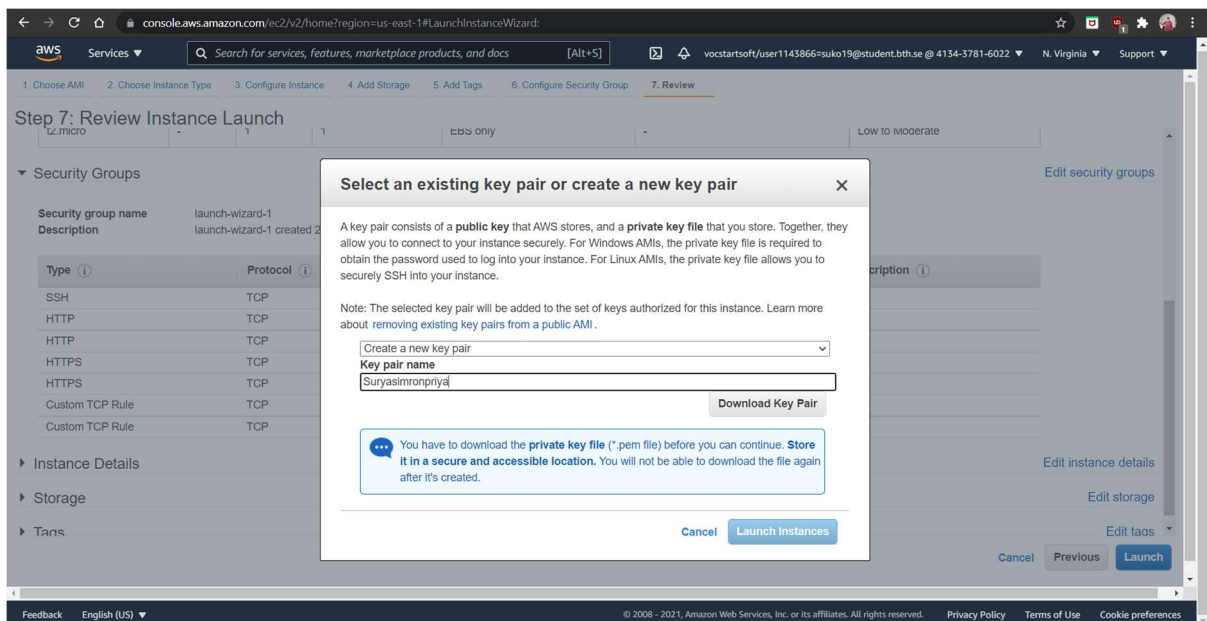
Description: launch-wizard-1 created 2021-06-10T19:52:19.432+02:00

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
HTTP	TCP	80	Custom 0.0.0.0/0, ::0	e.g. SSH for Admin Desktop
HTTPS	TCP	443	Custom 0.0.0.0/0, ::0	e.g. SSH for Admin Desktop
Custom TCP f	TCP	0	Anywhere 0.0.0.0/0, ::0	e.g. SSH for Admin Desktop

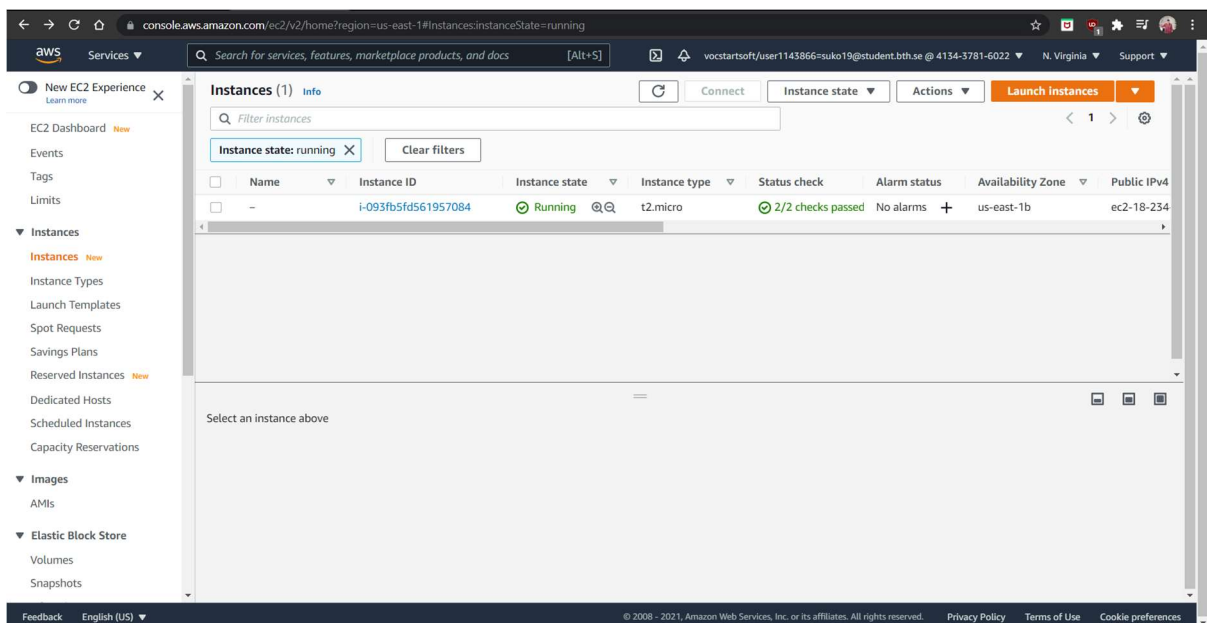
Add Rule

**Warning**  
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

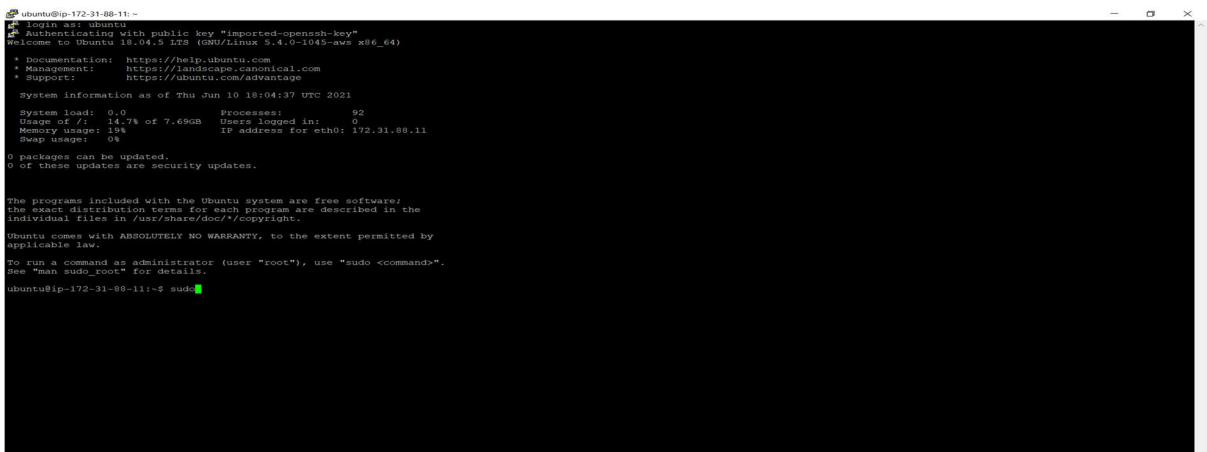
Cancel Previous **Review and Launch**



f. Now the Instance is running.



g. Now the EC2 instance is connected to the server using PuTTY application.



## Hosting the Web page

To host the web page, a web server is required which is installed using the following commands

```
sudo apt-get update
```

```
sudo apt-get install apache2
```

Now the document root is accessed for the Apache Web server, where Apache navigates to find files that can be served.

```
cd var/www/html
```

In this directory the web page is created that needs to be deployed in the Apache server

```
sudo nano webpage.html
```

The file that is created in the document root will now be hosted in the Apache server. This web page can now be configured for autoscaling using the Amazon Web Services in order to enable functionality to handle any large traffic to access the data in the web page.

## Autoscaling of the web application created using AWS Platform.

1. To achieve auto scaling, an image for the instance is generated in the application is created.

Name	AMI Name	AMI ID	Source	Owner	Visibility	Status	Creation Date	Platform
surya	ami-0fd971b36461c2022	ami-0fd971b36461c2022	413437816022/s...	413437816022	Private	available	June 10, 2021 at 8:24:47 PM...	Other Linux

Property	Value
AMI ID	ami-0fd971b36461c2022
Owner	413437816022
Status	available
Creation date	June 10, 2021 at 8:24:47 PM UTC+2
AMI Name	surya
Source	413437816022/surya
State Reason	-
Platform details	Linux/UNIX

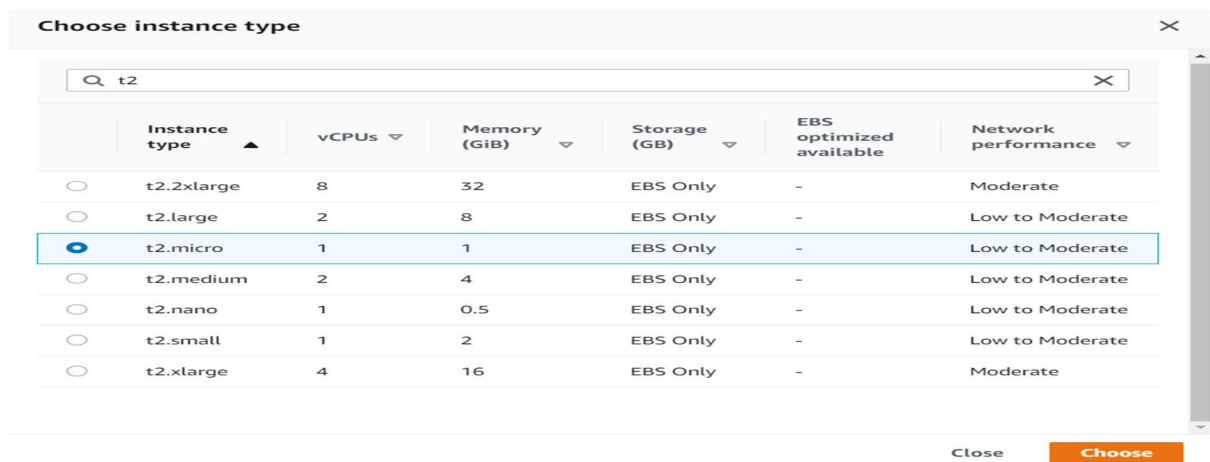
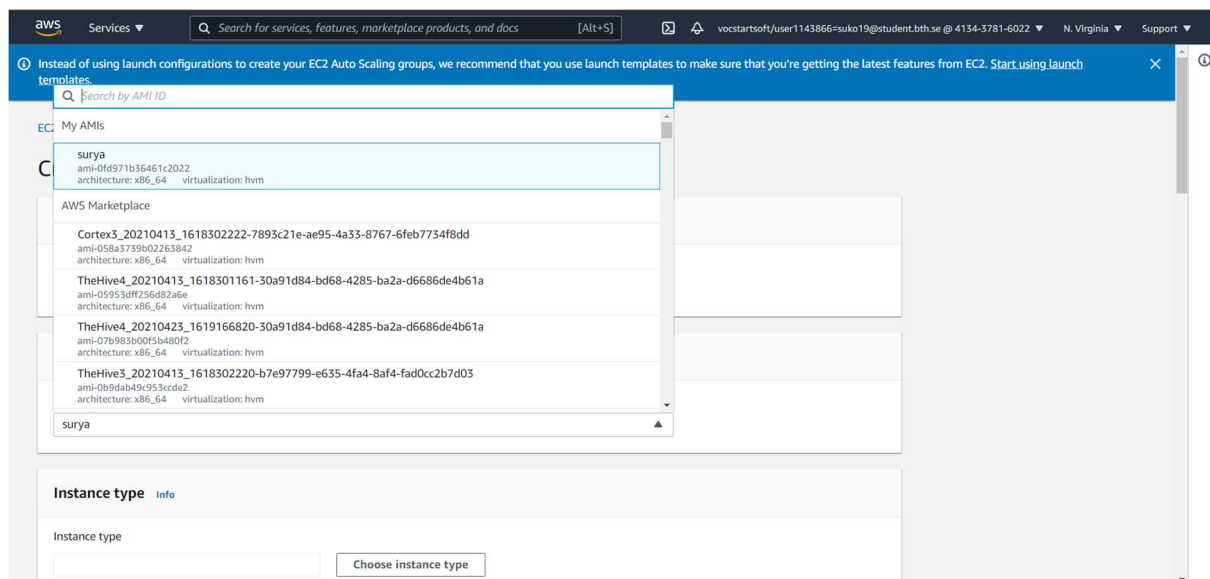
- 
2. A load balancer is configured for the application that is deployed. This is to handle the traffic for the application and achieve auto scaling. The default are selected for VPC and its subnets.

The screenshot shows the AWS Management Console interface for configuring a new Load Balancer. The top navigation bar includes the AWS logo, a search bar, and user information. The breadcrumb trail indicates the current step is '1. Configure Load Balancer'. The main heading is 'Step 1: Configure Load Balancer' with a sub-heading 'Basic Configuration'. Below this, a descriptive text explains the purpose of the configuration. The form includes fields for 'Name' (surya), 'Scheme' (Internet-facing), and 'IP address type' (IPv4). A 'Listeners' section shows a table with columns for 'Load Balancer Protocol' (HTTP) and 'Load Balancer Port' (80). An 'Add listener' button is present. The 'Availability Zones' section is partially visible at the bottom.

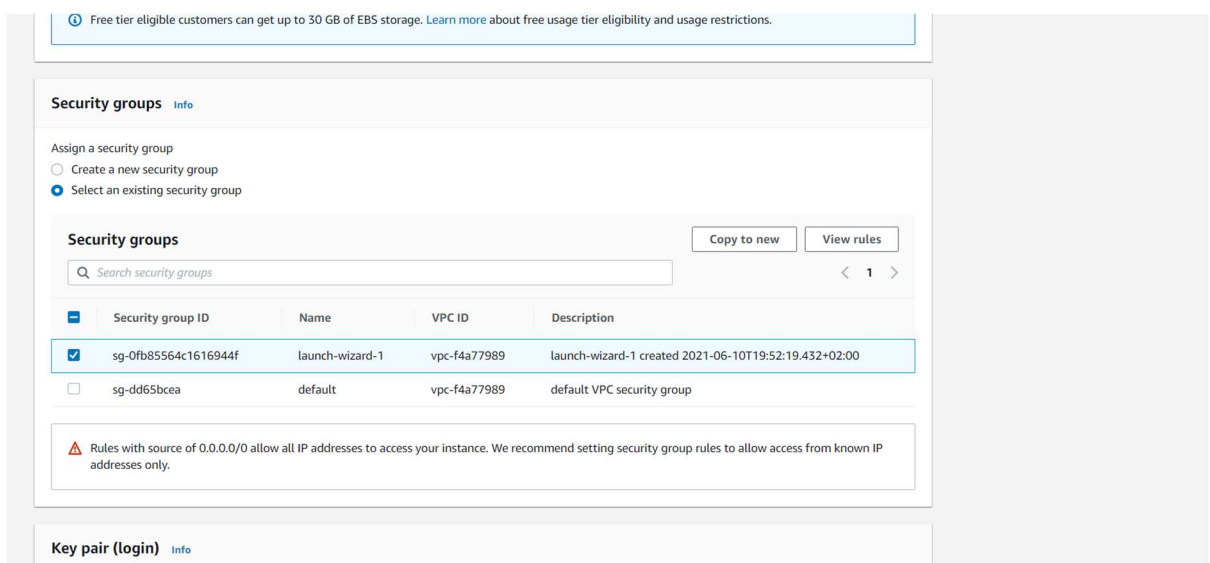
- 
- 
3. The security group is configured with HTTP and SSH inbound rules.

The screenshot shows the AWS Management Console interface for configuring a new Target Group. The breadcrumb trail indicates the current step is '4. Configure Routing'. The main heading is 'Step 4: Configure Routing' with a sub-heading 'Target group'. Below this, a descriptive text explains the purpose of the configuration. The form includes fields for 'Target group' (New target group), 'Name' (surya), 'Target type' (Instance), 'Protocol' (HTTP), 'Port' (80), and 'Protocol version' (HTTP1). A 'Cancel' button and a 'Next: Register Targets' button are visible at the bottom right.

- 
- 
- 
4. Now the load balancer is launched by generating a target group.
5. Further we create a configuration for the AMI generated for the instance using the t2.micro instance type.



- A security group is selected from the existing groups, or a new security group can be generated. Here, an existing security group is selected.





7. A key pair is required to create the launch configuration.

<input checked="" type="checkbox"/>	Security group ID	Name	VPC ID	Description
<input checked="" type="checkbox"/>	sg-0fb85564c1616944f	launch-wizard-1	vpc-f4a77989	launch-wizard-1 created 2021-06-10T19:52:19.432+02:00
<input type="checkbox"/>	sg-dd65bcea	default	vpc-f4a77989	default VPC security group

**Key pair (login)** [Info](#)

Key pair options

Choose an existing key pair

Existing key pair

Suryasimronpriya

☒ I acknowledge that I have access to the selected private key file (Suryasimronpriya.pem), and that without this file, I won't be able to log into my instance.

Cancel **Create launch configuration**

8. Utilising the launch configuration, we generate an autoscaling group. The launch configuration is selected and in the configuration settings the defaults are set for VPC and respective subnets are created.

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.

suryasimronpriya

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

**Launch configuration** [Info](#) [Switch to launch template](#)

Launch configuration

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

suryasimronpriya

Create a launch configuration <a href="#">Create a launch configuration</a>	AMI ID	Date created
Launch configuration	ami-0fd971b36461c2022	Thu Jun 10 2021 20:48:34 GMT+0200 (Central European Summer Time)
suryasimronpriya	Instance type	Key pair name
Security groups	t2.micro	-
sg-0fb85564c1616944f <a href="#">Link</a>		

Cancel **Next**



**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

surya | HTTP  
Application Load Balancer: surya

**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

9. To this autoscaling group that is generated and configured, the load balancer that has been generated previously will be attached.

**Configure settings** Info

Configure the settings below. Depending on whether you choose a launch template, these settings may include options to help you make optimal use of EC2 resources.

**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-f4a77989  
172.31.0.0/16 Default

Create a VPC

Subnets  
Select subnets

us-east-1a | subnet-7bff7d1d  
172.31.0.0/20 Default

us-east-1b | subnet-1e169f3f  
172.31.80.0/20 Default

Create a subnet

Cancel Previous Skip to review Next

10. Following, the group size is configured as desired by assigning minimum and maximum capacities for the group, and scaling policies are added.

**Configure group size and scaling policies** Info

Set the desired, minimum, and maximum capacity of your Auto Scaling group. You can optionally add a scaling policy to dynamically scale the number of instances in the group.

**Group size - optional** Info

Specify the size of the Auto Scaling group by changing the desired capacity. You can also specify minimum and maximum capacity limits. Your desired capacity must be within the limit range.

Desired capacity  
2

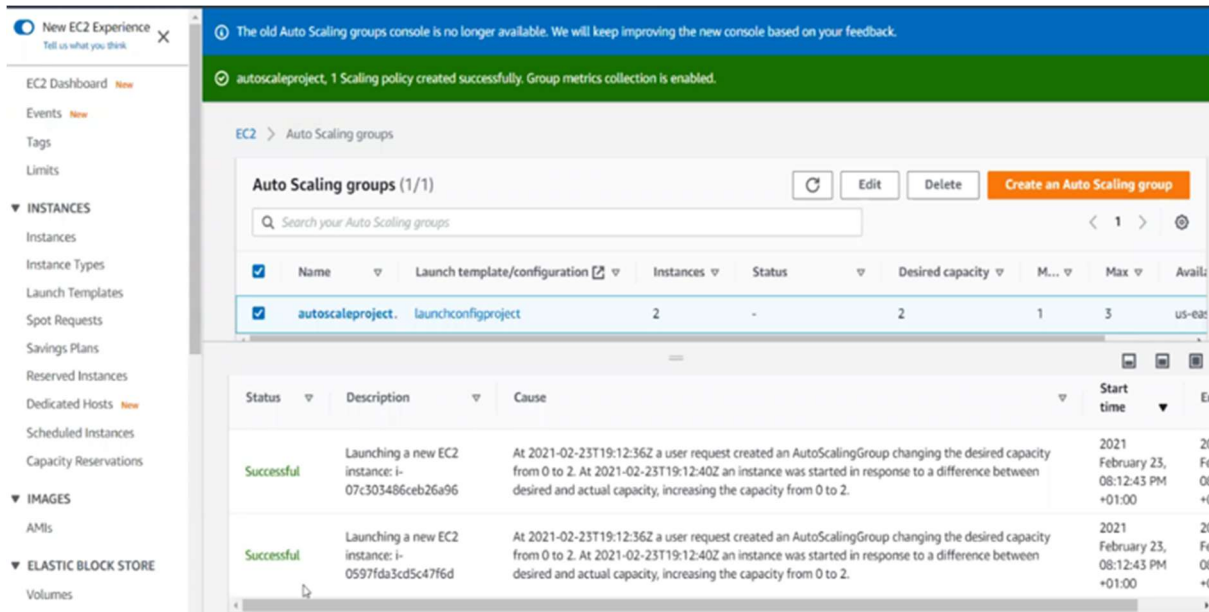
Minimum capacity  
1

Maximum capacity  
1

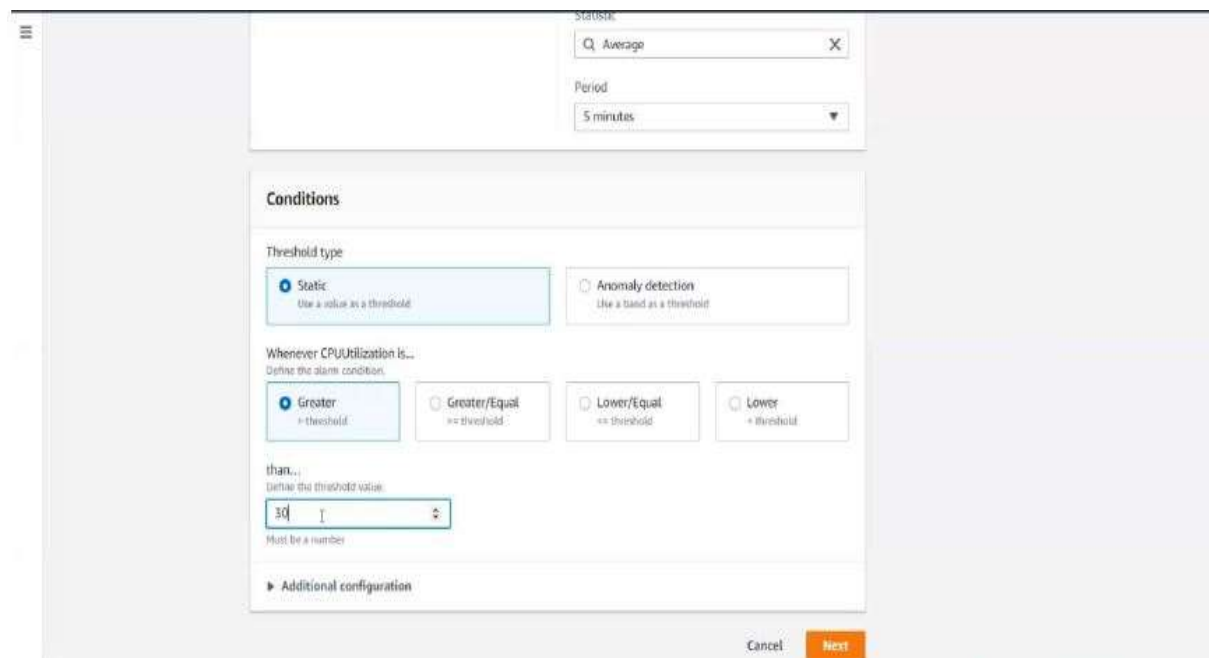
**Scaling policies - optional**

Choose whether to use a scaling policy to dynamically resize your Auto Scaling group to meet changes in demand. Info

11. Based on the number of configured instances, all the instances get launched and the webpage created is deployed in the instances. These can be found in the activity of the auto scaling group.



12. A threshold is set, and an alarm is created for the metric and the step scaling policy is generated. This will trigger when the data access in the web page reaches threshold.



13. The Apache bench is stressed to increase the requests to the web page which simulates the traffic to access the data in the web page.

```
Completed 50000 requests
Completed 100000 requests
Completed 150000 requests
Completed 200000 requests
Completed 250000 requests
```

## Verification

The web page created is deployed in the EC2 ubuntu instance. This web page can be accessed from the IPv4 address

***<http://ec2-3-93-212-172.compute-1.amazonaws.com/>***

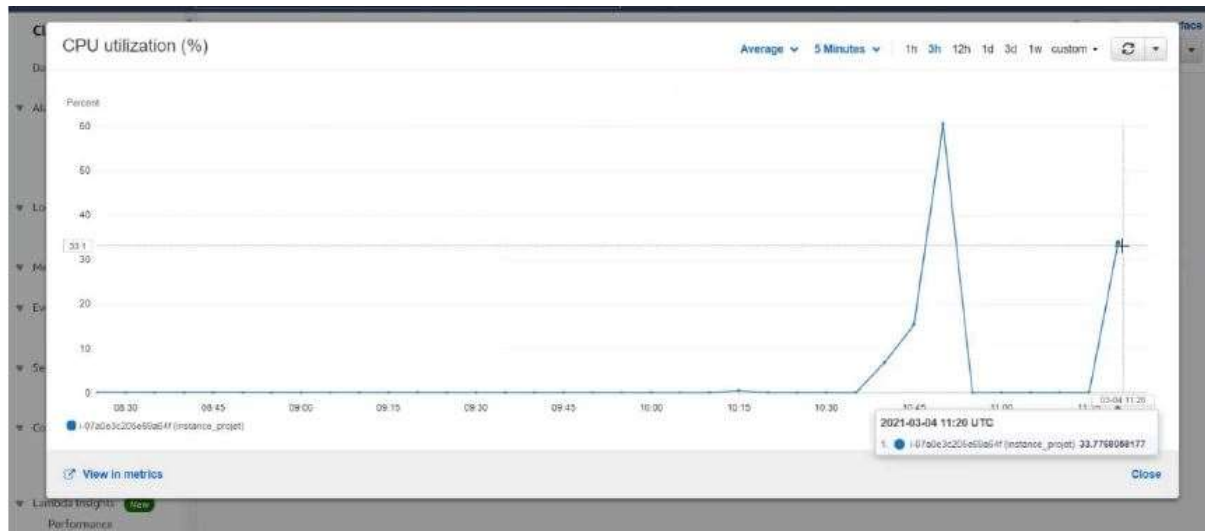
The autoscaling group with capacity 2 is launched with tracking policy. The Web page is deployed in both the instances and the instances are used for autoscaling. As soon the load reaches in an instance, the other instance gets initiated as the web page is also deployed in both the instances.



### The Tribute Page

Here we can dedicate any opinions, speeches or any other form of expression to near and dear. Your expressive words can be seen by many without any issues using this application.

The utilisation is increased as the stress is increased on the web page using the Apache test bench. The utilisation with the increase in the traffic can also be observed.



As soon as the stress is increased beyond the threshold level as configured in the step scaling policy, the scalability of the web page deployed is achieved.

Instances (1/3) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
autoscaling2	i-03ad0a2c9b613c7ff	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-18-210
autoscaling1	i-093fb5fd561957084	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	ec2-18-234
instance_pr...	i-035801a47e8f783dd	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	ec2-3-93-212-172

Instance summary Info

Instance ID i-035801a47e8f783dd (instance_project)	Public IPv4 address 3.93.212.172   <a href="#">open address</a>	Private IPv4 addresses 172.31.84.138
Instance state Running	Public IPv4 DNS ec2-3-93-212-172.compute-1.amazonaws.com   <a href="#">open address</a>	Private IPv4 DNS ip-172-31-84-138.ec2.internal
Instance type t2.micro	Elastic IP addresses	VPC ID

## Result

A web page has been created and deployed using the EC2 Ubuntu server in the Amazon Web Services. This is configured with the Apache Web Server, and Gunicorn WSGI is configured in-order to increase the functionality and to make the application run in the background. The web page with desired data is configured for autoscaling with two desired instances with a set threshold value for the traffic utilisation. The utilisation is for the Ubuntu server configured in the EC2 instance of the Amazon Web Service. The Web page is then subjected to stress by simulating large number of requests using the Apache test bench and when the threshold value is reached while stressing, the alarm will be set off, confining to the scaling policy, and thus achieving the autoscaling. Hence, the Web page created in the Amazon Web Services satisfies the autoscaling and scalability requirements with respect to the increased access for the data. The IPv4 link is <http://ec2-3-93-212-172.compute-1.amazonaws.com/>