



## SRI KRISHNA COLLEGE OF TECHNOLOGY

(An Autonomous Institution)

Approved by AICTE | Affiliated to Anna University Chennai|

Accredited by NBA - AICTE| Accredited by NAAC with 'A' Grade

KOVAIPUDUR, COIMBATORE 641042



### **DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

#### **22SB401 – Managing Cloud and Containerization**

**REGISTER NUMBER:**

Certified that this is the Bonafide record of work done by

Mr / Ms..... of the **Fourth Semester** .....

during the academic year **2023 – 2024 (EVEN SEMESTER)**.

**FACULTY IN-CHARGE**

**HOD**

Submitted for the End Semester Practical Examination held  
on ..... at **SRI KRISHNA COLLEGE OF TECHNOLOGY**,  
Kovaipudur, Coimbatore-42.

**INTERNAL EXAMINER**

**EXTERNAL EXAMINER**



**SRI KRISHNA COLLEGE OF TECHNOLOGY**  
(AN AUTONOMOUS INSTITUTION)

**KOVAIPUDUR, COIMBATORE 641042**



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

<b>Name of Lab Course</b>	<b>22SB401 – Managing Cloud and Containerization</b>
<b>Semester &amp; Year</b>	<b>IV &amp; II</b>
<b>Name of the Student</b>	
<b>Class</b>	<b>BE (CSE)</b>
<b>Name of the Evaluator</b>	
<b>Marks scored out of 100</b>	

## Rubric for Evaluating Continuous Lab Assessments

S.No	Components	Explanation of the Components	Excellent (>90% - <100%)	Good (>70% - <89%)	Average (>60% - <69%)	Below Average/poor (<50%)
1	<b>Aim, Algorithm, Flowchart (optional)</b> 20 marks	<ul style="list-style-type: none"> <li>• Level of understanding of the specific problem of the experiment.</li> <li>• Always start with the phrase like "the objective of this experiment is to".</li> <li>• Competence of the students to come up with step by step procedure to build the algorithm.</li> <li>• How well the student is able to represent the solution to the problem in a pictorial way.</li> </ul>	<ul style="list-style-type: none"> <li>• Student is able to understand the problem domain in an extremely good manner.</li> <li>• Capable of writing step by step procedure of the problem domain precisely.</li> <li>• Having enough skill to express the problem in a neat pictorial representation with appropriate symbols.</li> </ul> <p>(if applicable)</p>	<ul style="list-style-type: none"> <li>• Student is capable of expressing the problem domain in a good manner.</li> <li>• Able to write the step by step procedure in a simple way but not accurately.</li> <li>• Able to express the problem domain in a clear picture but not in a crispy way.(using more symbols) (if applicable)</li> </ul>	<ul style="list-style-type: none"> <li>• Student is able to understand the problem domain in a satisfactory manner.</li> <li>• Capable to write the step by step procedure but not exactly to the domain.</li> <li>• Able to provide a pictorial representation but not upto the standard specifications. (if applicable)</li> </ul>	<ul style="list-style-type: none"> <li>• Not capable of understanding the problem domain.</li> <li>• Not able to express the step by step procedure for the specified problem domain.</li> <li>• Unaware of the symbols and representations to express the problem domain in a required format. (if applicable)</li> </ul>

2	<b>Coding and Commands</b> 30 marks	<ul style="list-style-type: none"> <li>The ability of the students to write a Standard Set of instructions to execute/represent the problem domain. (including comment lines, detailed explanation of the particular instruction)</li> <li>How appropriately the syntax and semantics is written by the student.</li> </ul>	<ul style="list-style-type: none"> <li>Student is capable of writing the instructions syntactically and semantically in a precise manner.</li> <li>Able to write the coding optimistically and analyze the complexity.</li> <li>Able to use very apt terms and methodology.</li> </ul>	<ul style="list-style-type: none"> <li>Able to write the coding part based on syntax and semantics but not with the optimization.</li> <li>Complexity of the problem domain is considered.</li> <li>Able to use relevant terms and methodology.</li> </ul>	<ul style="list-style-type: none"> <li>Able to write the instructions but only with the help of Teacher.</li> <li>Complexity of the problem domain is considered but not upto the level.</li> <li>Some terms and methodology used.</li> </ul>	<ul style="list-style-type: none"> <li>Student is not capable of writing the coding part.</li> <li>Wrong usage of syntax and semantics.</li> <li>No usage of appropriate methodology.</li> </ul>
3	<b>Compilation and Debugging</b> 30 marks	<ul style="list-style-type: none"> <li>The skill of using appropriate inputs and outputs.</li> <li>How professionally the compilation procedure is done.</li> <li>The efficiency of the student in correcting the errors present in the coding.</li> </ul>	<ul style="list-style-type: none"> <li>Student is capable of compiling the program in a perfect manner without errors.</li> <li>Able to use appropriate syntax/ semantics and debug with one or more nil errors.</li> </ul>	<ul style="list-style-type: none"> <li>Student is capable of compiling the program but with few errors.</li> <li>Able to use appropriate syntax/ semantics and debug with one or more warnings.</li> </ul>	<ul style="list-style-type: none"> <li>Student is liable to use incorrect syntaxes due to lack of knowledge.</li> <li>Capable of correcting the errors but only with the help of Teacher.</li> </ul>	<ul style="list-style-type: none"> <li>Student is unaware of the compilation procedures.</li> <li>Not capable of producing the results.</li> </ul>
4	<b>Execution and Results</b> 10 marks	<ul style="list-style-type: none"> <li>Obtaining the exact solution to the given problem and accurate result to the aim without errors.</li> <li>Inclusion of generalization of the problem.</li> </ul>	<ul style="list-style-type: none"> <li>Student is capable of producing exact results for various types of inputs.</li> <li>Able to produce a generalized solution to the problem.</li> </ul>	<ul style="list-style-type: none"> <li>Student is capable of producing results only for particular inputs.</li> <li>Able to try some part to generalize.</li> </ul>	<ul style="list-style-type: none"> <li>Student is obtaining the solution with compile or run time errors and able to rectify it with the help of a teacher.</li> <li>No choice of generalization.</li> </ul>	<ul style="list-style-type: none"> <li>Student is not able to furnish a related solution to the problem.</li> <li>Lack of knowledge in giving appropriate inputs.</li> <li>No choice of generalization.</li> </ul>

5	<b>Documentation-Explanation of the experiment/Viva</b> 10 marks	<ul style="list-style-type: none"> <li>• How provisionally the Documentation and presentation of the components from 1-4 is done.</li> <li>• The ability of the student in explaining and demonstrating the solved problem.</li> </ul>	<ul style="list-style-type: none"> <li>• Student is able to bring out a document in an efficient manner by including all the components from 1 to 4.</li> <li>• Able to give all the explanations for the solved problem.</li> </ul>	<ul style="list-style-type: none"> <li>• Student is able to produce the document with some components but not in a prescribed format.</li> <li>• Able to explain the problem but not clearly.</li> </ul>	<ul style="list-style-type: none"> <li>• Student is not able to produce the document in a prescribed format.</li> <li>• Able to explain the problem but irrelevant to the concept.</li> </ul>	<ul style="list-style-type: none"> <li>• Student is not able to produce the document even with the help of teacher.</li> <li>• Unable to explain the problem.</li> </ul>
---	---	--	--	--	---	--

**FACULTY IN-CHARGE**





**SRI KRISHNA COLLEGE OF TECHNOLOGY  
(AN AUTONOMOUS INSTITUTION)**

**KOVAIPUDUR, COIMBATORE 641042**



<b>EXPERIMENT NO.</b>	:
<b>TITLE OF EXPERIMENT</b>	:
<b>DATE OF EXPERIMENT</b>	:

**EVALUATION BY FACULTY MEMBER (BASED ON RUBRICS)**

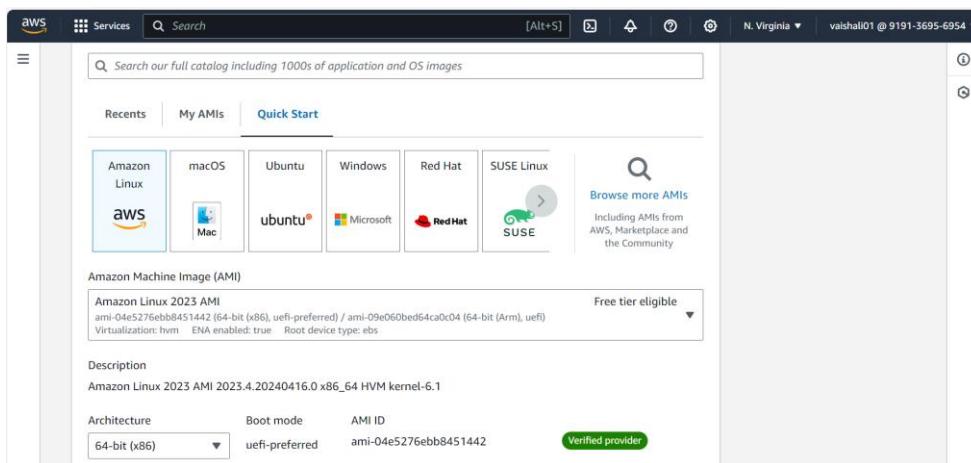
<b>CRITERIA</b>	<b>MAXIMUM MARKS</b>	<b>MARKS SCORED BY STUDENT</b>
Aim, Algorithm, Flowchart (optional)	20	
Coding and Commands	30	
Compilation and Debugging	30	
Execution and Result	10	
Documentation -Explanation of the experiment/ Viva	10	
<b>TOTAL MARKS</b>	<b>100</b>	
Signature of the faculty		

# 1. Create an EC2 Instance in the us-east-1 region with the following requirements.

**AIM:**

**ALGORITHM:**

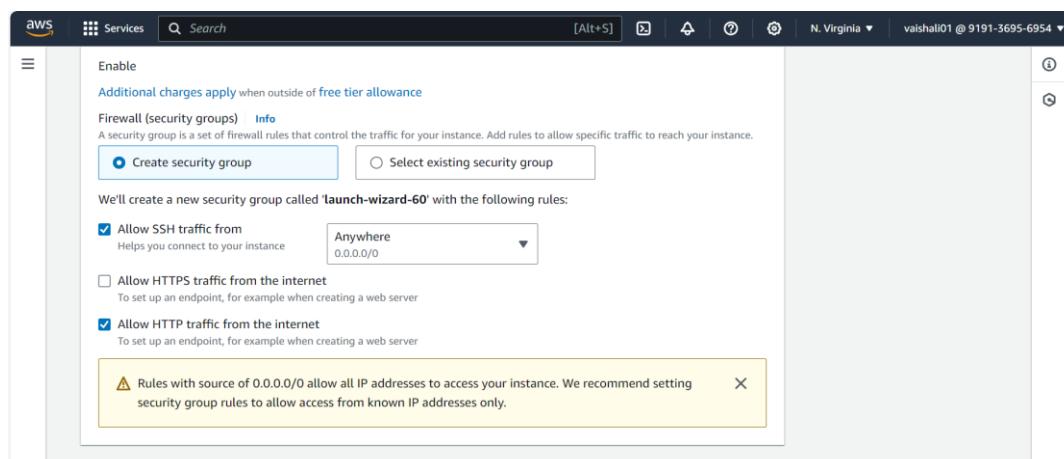
- Give the Name tag of both EC2 instance & keypair as "ec2usecase1"(Name).  
STEP 1: Search EC2 -> Change location to **us-east-1 N.Virginia** click Launch instance and set EC2 instance & Keypair name -> ec2usecase1".
- EC2 instance AMI should be "Amazon Linux 2".



- Allow SSH traffic for taking putty remote connection.

- Allow HTTP traffic from the internet for reaching website requests.

**STEP 2 :** Click on check box Allow SSH traffic from and Click on check box Allow HTTP traffic from the internet.



**Note:** The EC2 Instance type must be either t2.micro or t3.micro

The screenshot shows the AWS EC2 instance creation process. On the left, a sidebar lists services like EC2 Dashboard, EC2 Global View, Events, and Instances. The main area displays the 'Amazon Linux 2023 AMI' configuration. Key details include:

- AMI:** ami-04e5276ebb8451442 (64-bit (x86), uefi-preferred) / ami-09e060bed64ca0c04 (64-bit (Arm), uefi)
- Architecture:** 64-bit (x86)
- Boot mode:** uefi-preferred
- AMI ID:** ami-04e5276ebb8451442
- Virtualization:** hvm
- ENAs enabled:** true
- Root device type:** ebs

The 'Free tier eligible' status is indicated. Below this, the 'Instance type' section shows a t2.micro instance selected. Other options like t2.small and t2.medium are also listed. The 'Launch instance' button is prominently displayed at the bottom right.

STEP 3: Click Launch instance.

The screenshot shows the AWS EC2 Instances page. The left sidebar includes 'Instances' (selected), 'Images', and 'Elastic Block Store'. The main area displays a table of existing instances, with one row for 'ec2usecase1' (Instance ID: i-0179efa744d5f91c8, State: Running, Type: t2.micro). A modal window titled 'Select an instance' is open over the table, indicating the user has selected the instance for launching.

**Result:**



**SRI KRISHNA COLLEGE OF TECHNOLOGY  
(AN AUTONOMOUS INSTITUTION)**

**KOVAIPUDUR, COIMBATORE 641042**



<b>EXPERIMENT NO.</b>	:
<b>TITLE OF EXPERIMENT</b>	:
<b>DATE OF EXPERIMENT</b>	:

**EVALUATION BY FACULTY MEMBER (BASED ON RUBRICS)**

<b>CRITERIA</b>	<b>MAXIMUM MARKS</b>	<b>MARKS SCORED BY STUDENT</b>
Aim, Algorithm, Flowchart (optional)	20	
Coding and Commands	30	
Compilation and Debugging	30	
Execution and Result	10	
Documentation -Explanation of the experiment/ Viva	10	
<b>TOTAL MARKS</b>	<b>100</b>	
Signature of the faculty		

## 2. Create a EC2 windows instance in the region (us-east-1) with the following requirements.

AIM:

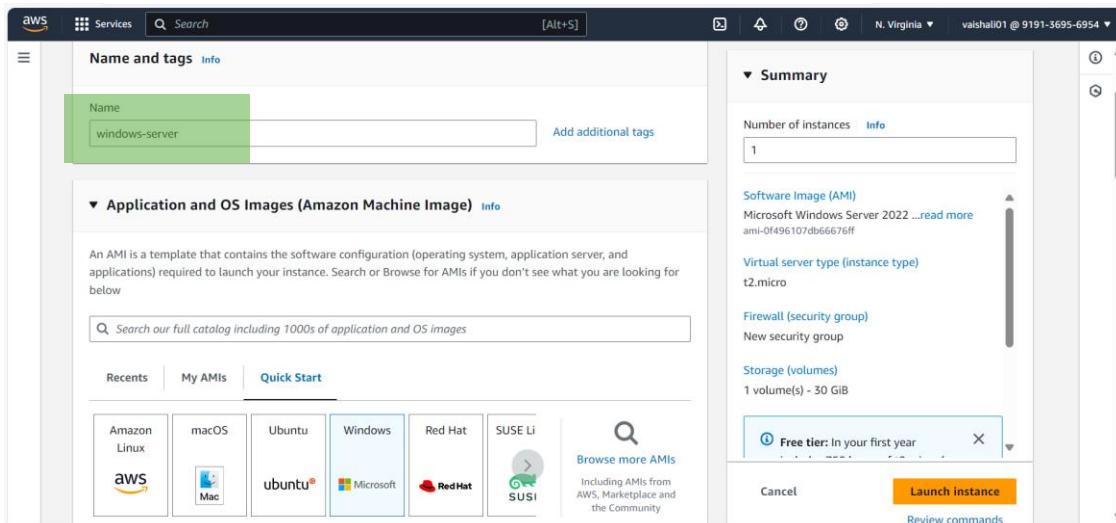
### ALOGRITHM:

- Give the name tag of both ec2 instance and key pair as “windows-server”.

STEP 1: Search EC2 -> Change location to **us-east-1 N.Virginia** click Launch instance and set EC2 instance & Keypair name -> windows-server.

- EC2 instance AMI should be "Microsoft Windows Server".

STEP 2: Select from Quick start as Microsoft Windows Server.

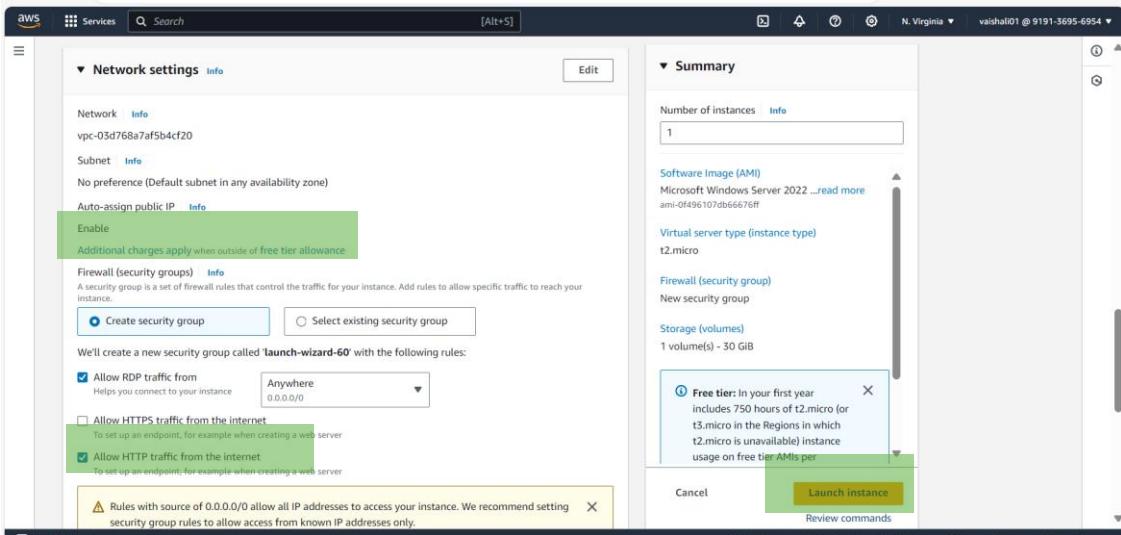


- The "Auto-Assign public IP" should be enabled.

- "Port 80" should be exposed in the security group.

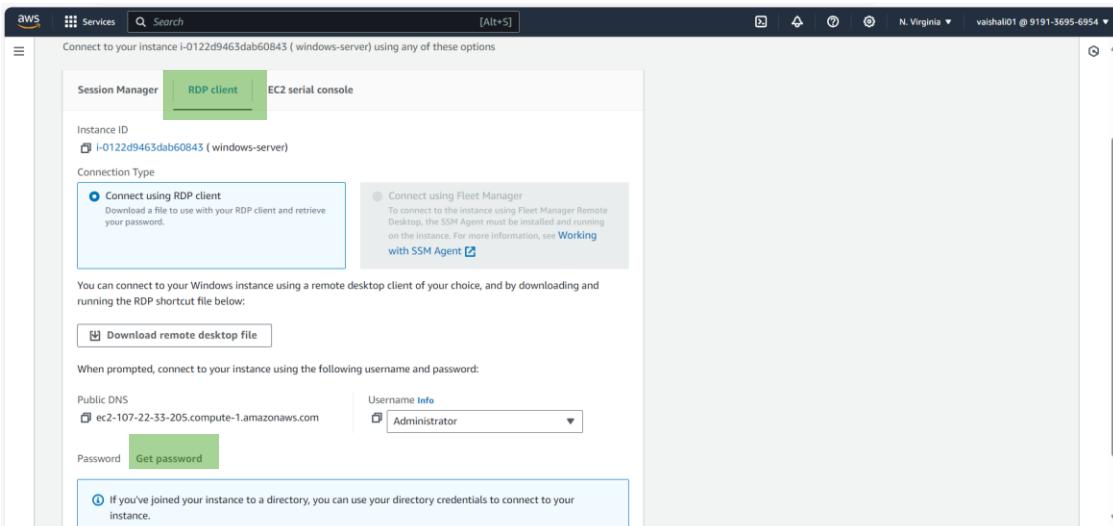
STEP 3: In Network setting check where Auto-assign public IP is enabled.

STEP 4: Click the check box Allow HTTP traffic from the internet to set up an endpoint, for example when creating a web server.

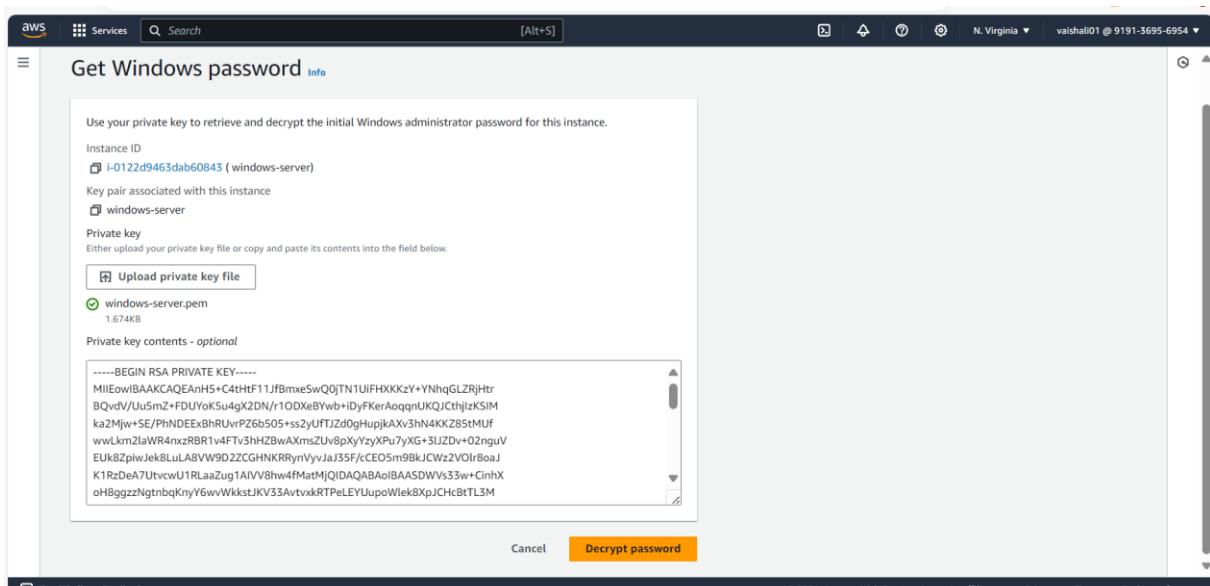


- Login into window server using RDP.

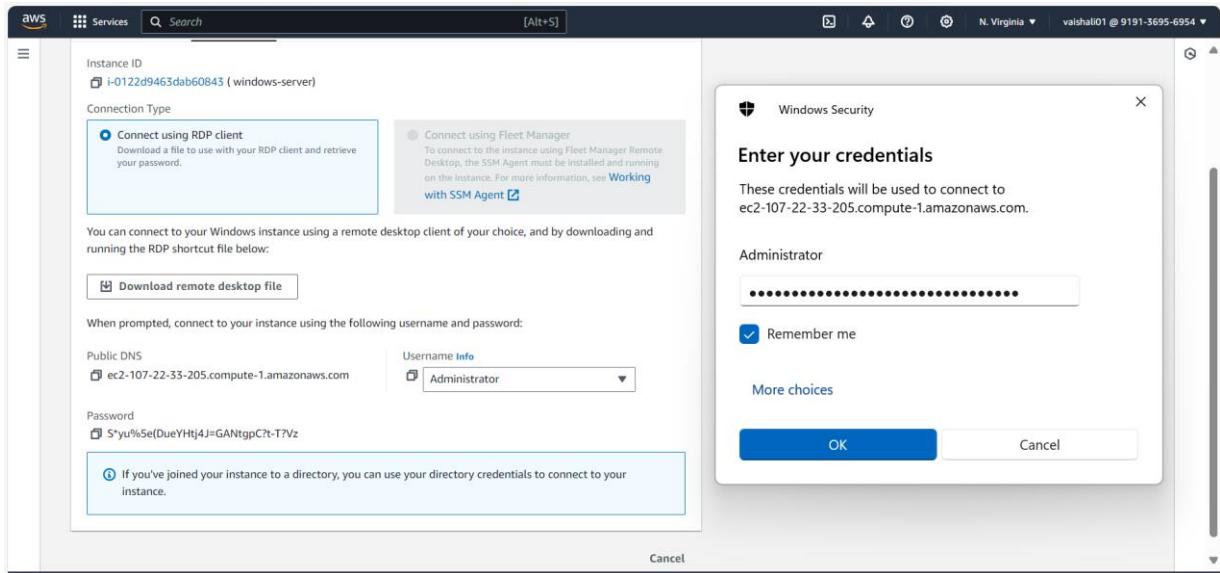
STEP 4: Click on Launch instance -> Click on connect -> Choose RDP client -> click download remote desktop file.



STEP 5: Click on get password and Upload Private key from file ,Now click the Decrypt password.

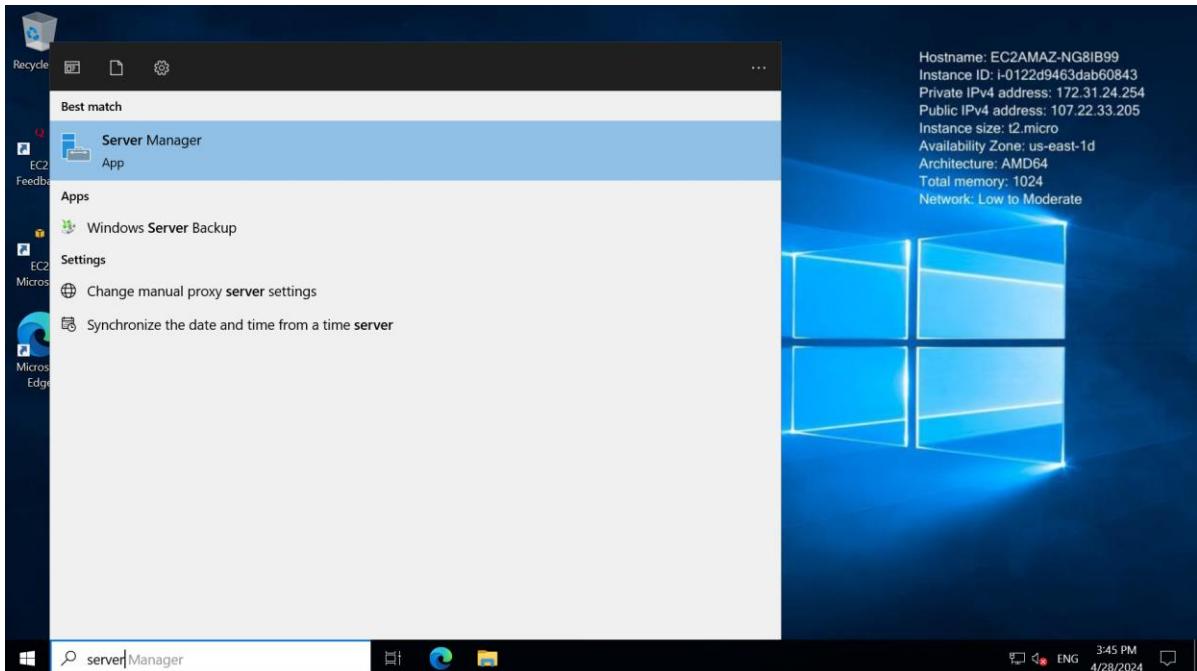


STEP 6: Open the EC2 machine which is Download, and copy the password and paste in the EC2 machine and click Okay and proceed with YES.



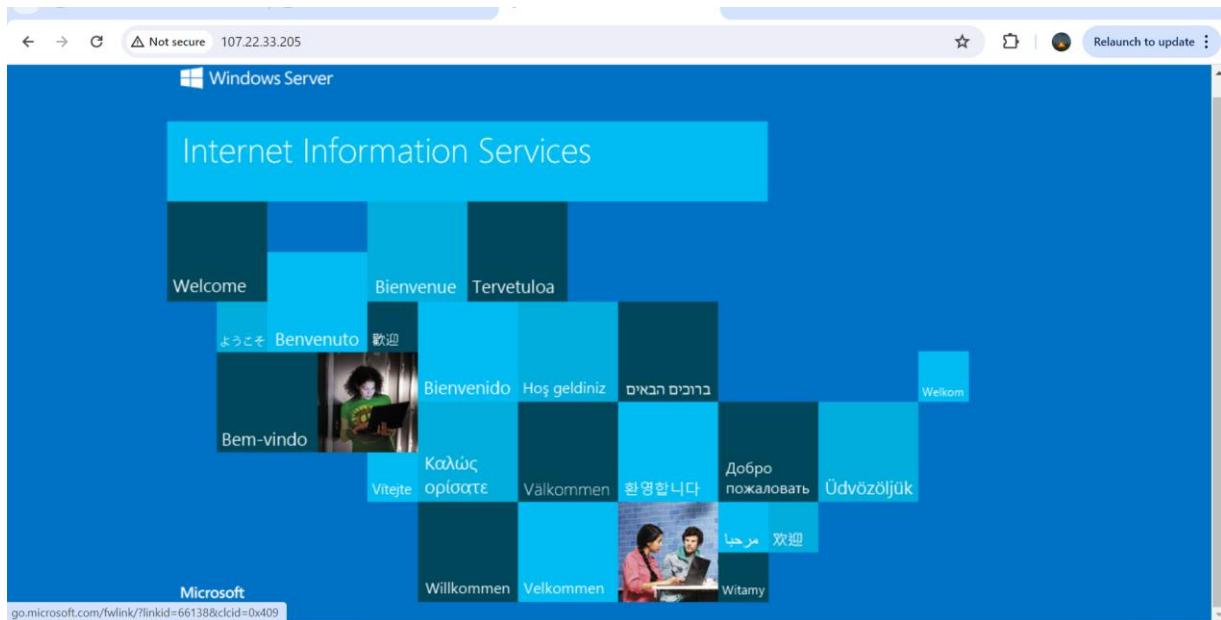
STEP 7: In EC2 machine open Server manager and add roles and features -> click Next , Installation type ->click next, Server Selection -> click Next, Server Roles -> Click Next, Features -> IIS hostable web core.

STEP 8: Confirmation -> click install, Result -> Installation succeed.



STEP 8: Go back to the EC2 instance and select the Ip address in details Public IPv4 address and paste it in the browser.

- Enable IIS and show the IIS page output with the help of public IP.
- Ensure that the "WebService" is running.



**Note:** The EC2 Instance type must be either t2.micro or t3.micro

**RESULT:**



**SRI KRISHNA COLLEGE OF TECHNOLOGY  
(AN AUTONOMOUS INSTITUTION)**

**KOVAIPUDUR, COIMBATORE 641042**



<b>EXPERIMENT NO.</b>	:
<b>TITLE OF EXPERIMENT</b>	:
<b>DATE OF EXPERIMENT</b>	:

**EVALUATION BY FACULTY MEMBER (BASED ON RUBRICS)**

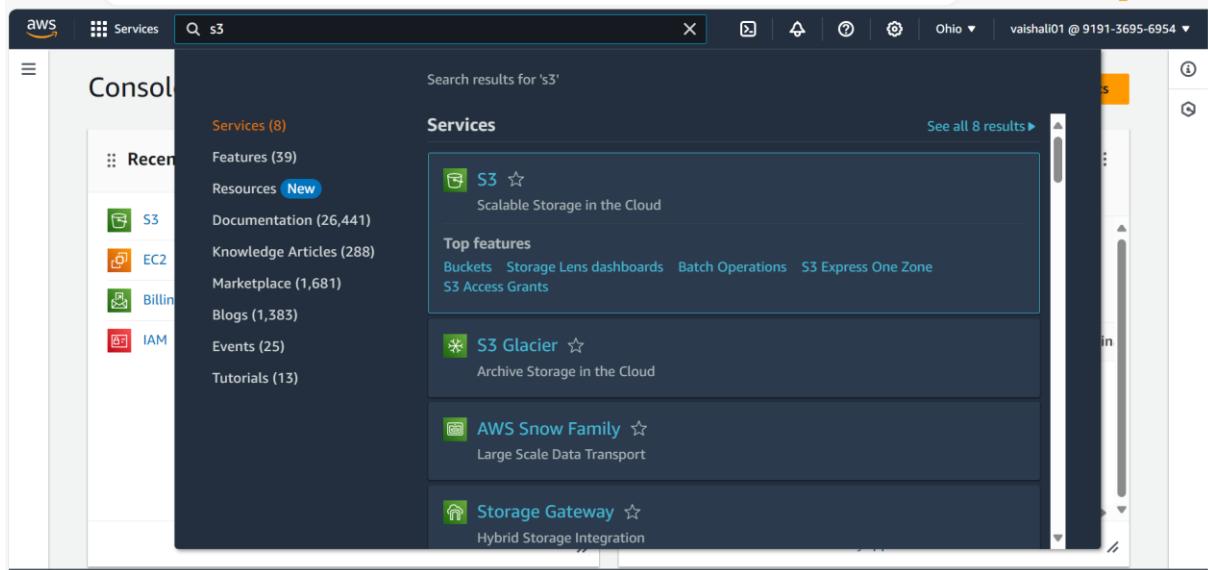
<b>CRITERIA</b>	<b>MAXIMUM MARKS</b>	<b>MARKS SCORED BY STUDENT</b>
Aim, Algorithm, Flowchart (optional)	20	
Coding and Commands	30	
Compilation and Debugging	30	
Execution and Result	10	
Documentation -Explanation of the experiment/ Viva	10	
<b>TOTAL MARKS</b>	<b>100</b>	
Signature of the faculty		

### 3. Create an S3 bucket with the following requirements

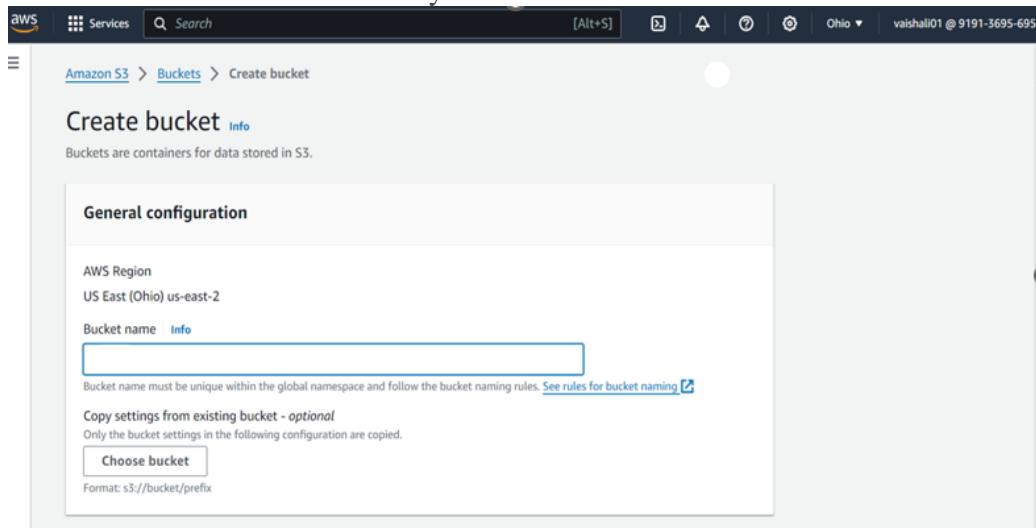
AIM:

#### ALGORITHM:

- Create an S3 bucket in the Asia location of 'us-east-2'(ohio).



- Create a folder in the name of 'myfolder' in the created bucket.



STEP 2: Create a folder 'imagefolder01', In Object Ownership click ->ACLs enabled and disable the **Block all public access**

#### Block public access (bucket settings)

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)



Account settings for Block Public Access are currently turned on

Account settings for Block Public Access that are enabled apply even if they are disabled for this bucket.

##### **Block all public access**

Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another

###### **Block public access to buckets and objects granted through new access control lists (ACLs)**

S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.

###### **Block public access to buckets and objects granted through any access control lists (ACLs)**

S3 will ignore all ACLs that grant public access to buckets and objects.

###### **Block public access to buckets and objects granted through new public bucket or access point policies**

S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.

###### **Block public and cross-account access to buckets and objects through any public bucket or access point policies**

S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and

- Upload a file named 'mybiodata.pdf' under the folder 'imagefolder01'.

STEP 3: Click the 'imagefolder01' bucket -> click UPLOAD and Add files-> click upload

The screenshot shows the AWS S3 'Upload' interface. At the top, there's a navigation bar with the AWS logo, 'Services' dropdown, and a search bar. On the right, there are settings and a location dropdown set to 'Ohio'. Below the navigation is a large 'Upload' button with an 'Info' link. A message says 'Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. [Learn more](#)'. A central area has a placeholder 'Drag and drop files and folders you want to upload here, or choose Add files or Add folder.' Below this is a 'Files and folders' table showing one item: 'person.png' (3.3 KB). The table includes columns for Name, Folder, and Type. At the bottom, there's a 'Destination' section with an 'Info' link.

- Make the object 'mybiodata.pdf' file accessible to everyone(publicly).

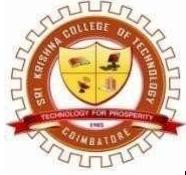
STEP 5: In Files and folder click-> imagefolder01 and click the READ check box.

## RESULT:



**SRI KRISHNA COLLEGE OF TECHNOLOGY  
(AN AUTONOMOUS INSTITUTION)**

**KOVAIPUDUR, COIMBATORE 641042**



<b>EXPERIMENT NO.</b>	:
<b>TITLE OF EXPERIMENT</b>	:
<b>DATE OF EXPERIMENT</b>	:

**EVALUATION BY FACULTY MEMBER (BASED ON RUBRICS)**

<b>CRITERIA</b>	<b>MAXIMUM MARKS</b>	<b>MARKS SCORED BY STUDENT</b>
Aim, Algorithm, Flowchart (optional)	20	
Coding and Commands	30	
Compilation and Debugging	30	
Execution and Result	10	
Documentation -Explanation of the experiment/ Viva	10	
<b>TOTAL MARKS</b>	<b>100</b>	
Signature of the faculty		

4. Create an IAM group called 'Network-L1-Team' with 'AmazonVPCReadOnlyAccess' and 'AWSNetworkManagerReadOnlyAccess' policies, then add an IAM user called 'Network-L1-User1' to the group.

**AIM:**

- The name of the IAM group should be 'Network-L1-Team'.

- The name of the IAM user should be 'Network-L1-User1'.

STEP 1: Search IAM services and Create a IAM group 'Network-L1-Team'.

Policy name	Type	Used as	Description
AdministratorAccess	AWS managed - job function	Permissions policy (2)	Provides full access to AWS services an...
AdministratorAccess...	AWS managed	None	Grants account administrative permis...
AdministratorAccess-A...	AWS managed	None	Grants account administrative permis...
AlexaForBusinessDev...	AWS managed	None	Provide device setup access to AlexaFo...
AlexaForBusinessFullA...	AWS managed	None	Grants full access to AlexaForBusiness ...
AlexaForBusinessGate...	AWS managed	None	Provide gateway execution access to A...
AlexaForBusinessLifes...	AWS managed	None	Provide access to Lifesize AVS devices
AlexaForBusinessPoly...	AWS managed	None	Provide access to Poly AVS devices
AlexaForBusinessRead...	AWS managed	None	Provide read only access to AlexaForB...
AmazonAPIGatewayA...	AWS managed	None	Provides full access to create/edit/delete...
AmazonAPIGatewayIn...	AWS managed	None	Provides full access to invoke APIs in A...
AmazonAPIGatewayP...	AWS managed	None	Allows API Gateway to push logs to us...

STEP 2: Select the respected permissions

- The 'AmazonVPCReadOnlyAccess' policy should be attached.
- The 'AWSNetworkManagerReadOnlyAccess' policy should be attached.

## STEP 3: Check the Permission policy.

The screenshot shows the AWS Identity and Access Management (IAM) console. The left sidebar is collapsed. The main area displays the 'User groups' section under 'Access management'. A specific user group, 'Network-L1-Team', is selected. The 'Permissions' tab is active, showing two managed policies attached to the group: 'AmazonVPCReadOnlyAccess' and 'AWSNetworkManagerReadOnlyAccess'. Both policies are AWS managed and have one attached entity.

Policy name	Type	Attached entities
AmazonVPCReadOnlyAccess	AWS managed	1
AWSNetworkManagerReadOnlyAccess	AWS managed	1

**RESULT:**



**SRI KRISHNA COLLEGE OF TECHNOLOGY  
(AN AUTONOMOUS INSTITUTION)**

**KOVAIPUDUR, COIMBATORE 641042**



<b>EXPERIMENT NO.</b>	:
<b>TITLE OF EXPERIMENT</b>	:
<b>DATE OF EXPERIMENT</b>	:

**EVALUATION BY FACULTY MEMBER (BASED ON RUBRICS)**

<b>CRITERIA</b>	<b>MAXIMUM MARKS</b>	<b>MARKS SCORED BY STUDENT</b>
Aim, Algorithm, Flowchart (optional)	20	
Coding and Commands	30	
Compilation and Debugging	30	
Execution and Result	10	
Documentation -Explanation of the experiment/ Viva	10	
<b>TOTAL MARKS</b>	<b>100</b>	
Signature of the faculty		

## 5. Configure an AWS Application Load Balancer to distribute traffic to EC2 instances across multiple availability zones within a single region "N.Virginia"

**AIM:**

### ALOGRITHM:

- Create an EC2 instance with the name tag "serverinstance1" in the "us-east-1a" availability zone.

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with options like EC2 Dashboard, EC2 Global View, Events, and Instances. Under Instances, there are sub-options for Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity, and Reservations. Below that is another section for Images. The main area shows a table with one row for the instance i-07704205a58d68d66, which is running an Ubuntu Docker AMI. The instance has a public IPv4 address of 18.212.81.15 and a private IPv4 address of 172.31.16.31. The status bar at the bottom indicates "Public IPv4 DNS".

The screenshot shows the "Launch an instance" wizard. It starts with a summary step where you can choose the number of instances (1), the software image (Amazon Linux 2023 AMI 2023.4.2...), the virtual server type (t2.micro), and storage (1 volume(s) - 8 GiB). Then it goes through a "Name and tags" step where you can enter a name (serverinstance1) and add tags. Finally, it reaches an "Application and OS Images (Amazon Machine Image)" step where you can search for AMIs. At the bottom right is a large orange "Launch instance" button.

- Create another EC2 instance with the name tag "serverinstance2" in the "us-east-1b" availability zone

**Name and tags**

Name: serverinstance2

**Application and OS Images (Amazon Machine Image)**

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

Quick Start AMIs, My AMIs, AWS Marketplace AMIs, Community AMIs

- Configure a http website in both the instances and allow http traffic (Port: 80) in the security group

**Inbound Security Group Rules**

**Security group rule 1 (TCP, 22, 0.0.0.0/0)**

Type: ssh, Protocol: TCP, Port range: 22  
Source type: Anywhere, Description: e.g. SSH for admin desktop

**Security group rule 2 (TCP, 80, 0.0.0.0/0)**

Type: HTTP, Protocol: TCP, Port range: 80  
Source type: Anywhere, Description: e.g. SSH for admin desktop

Quick Start AMIs, My AMIs, AWS Marketplace AMIs, Community AMIs

- Create a Target group with the name "servertargets"

**Application Load Balancer**

- Offers the flexibility for a Network Load Balancer to accept and route TCP requests within a specific VPC.
- Facilitates using static IP addresses and PrivateLink with an Application Load Balancer.

**Target group name**: servertargets

**Protocol : Port**: HTTP : 80

**IP address type**: IPv4

- Register the above created two instances in the "servertargets" Target Group

**Available instances (2/2)**

Instance ID	Name	State	Security groups
i-0bc8658d801bb8724	serverinstance2	Running	launch-wizard-58
i-0dc7b92eb96518496	serverinstance1	Running	launch-wizard-57

**2 selected**

**Ports for the selected instances**  
Ports for routing traffic to the selected instances.

80  
1-65535 (separate multiple ports with commas)

**Include as pending below**

Add two instance in the target group

**Review targets**

**Targets (2)**

Instance ID	Name	Port	State	Security groups	Zone
i-0bc8658d801bb8724	serverinstance2	80	Running	launch-wizard-58	us-east-1b
i-0dc7b92eb96518496	serverinstance1	80	Running	launch-wizard-57	us-east-1a

**2 pending**

**Create target group**

Register the above created two instances in the "servertargets" Target Group

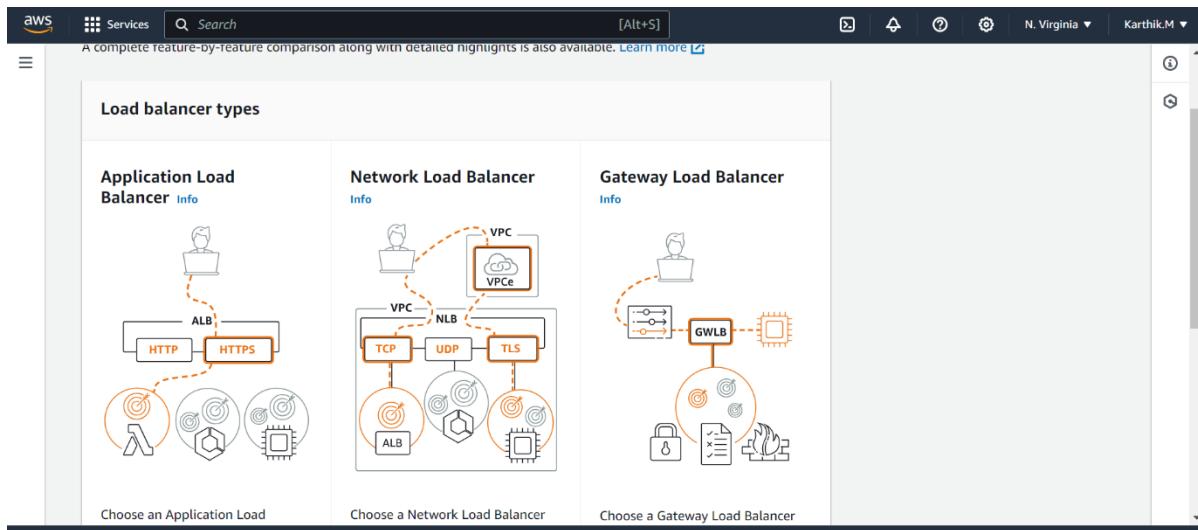
- Create Loadbalancer with the name "serverelb" by using "servertargets" Target Group

**Load balancers**

No load balancers  
You don't have any load balancers in us-east-1

**Create load balancer**

- Create Application load balancer



- Give a name for application load balancer

The screenshot shows the 'Create Application Load Balancer' wizard in the AWS Management Console. The current step is 'Basic configuration'.

**Load balancer name:** serverlb

**Scheme:** Info

- Select -> Target group in this load balancer

The screenshot shows the 'Listeners and routing' configuration for an Application Load Balancer. A single listener is defined:

- Listener HTTP:80**
- Protocol:** HTTP
- Port:** 80
- Default action:** Forward to **servertargets** (Target type: Instance, IPv4)
- Create target** button
- Listener tags - optional**: Add listener tag

- Create Load Balancer

The screenshot shows the AWS EC2 Load Balancers console. The left sidebar has a tree view with 'Instances' expanded, showing 'Instances', 'Instance Types', 'Launch Templates', 'Spot Requests', 'Savings Plans', 'Reserved Instances', 'Dedicated Hosts', 'Capacity', and 'Reservations'. A 'Create load balancer' button is visible at the top right of the main content area. The main content area displays a table titled 'Load balancers (1)'. The table has columns for Name, DNS name, State, VPC ID, and Availability. One row is shown, labeled 'serverlb' with a DNS name of 'serverlb-1896824553.us-east-1.elasticloadbalancing.amazonaws.com', a state of 'Provisioning..', a VPC ID of 'vpc-03d768a7af5b4cf20', and an availability of '3 Available'. A note at the bottom says '0 load balancers selected'.

**Note:** While creating any AWS resources as per question either in Terraform or console, if you see any error saying that the resource "already exists" then kindly delete the existing resource and create your new resource. If VPC limit exceeds also, then kindly delete the existing VPC and create your new VPC.

## RESULT:



**SRI KRISHNA COLLEGE OF TECHNOLOGY  
(AN AUTONOMOUS INSTITUTION)**

**KOVAIPUDUR, COIMBATORE 641042**



<b>EXPERIMENT NO.</b>	:
<b>TITLE OF EXPERIMENT</b>	:
<b>DATE OF EXPERIMENT</b>	:

**EVALUATION BY FACULTY MEMBER (BASED ON RUBRICS)**

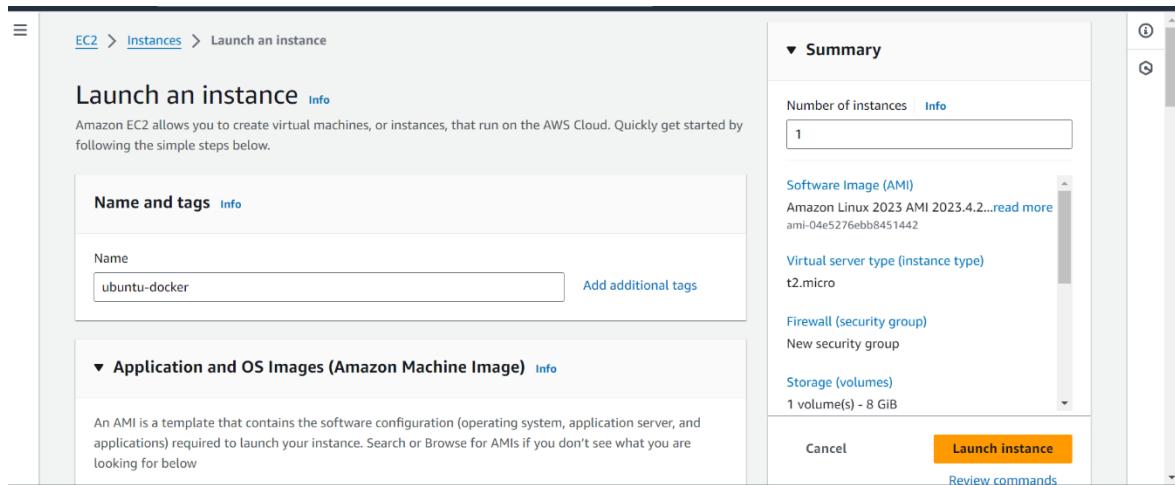
<b>CRITERIA</b>	<b>MAXIMUM MARKS</b>	<b>MARKS SCORED BY STUDENT</b>
Aim, Algorithm, Flowchart (optional)	20	
Coding and Commands	30	
Compilation and Debugging	30	
Execution and Result	10	
Documentation -Explanation of the experiment/ Viva	10	
<b>TOTAL MARKS</b>	<b>100</b>	
Signature of the faculty		

## 6. Create a docker container with the following requirements.

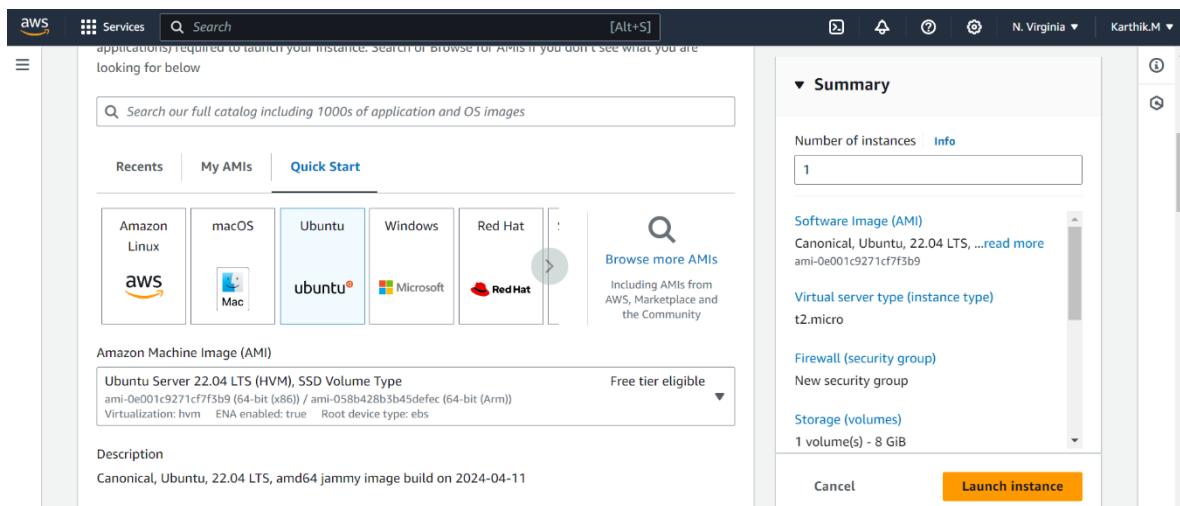
**AIM:**

**ALGORITHM:**

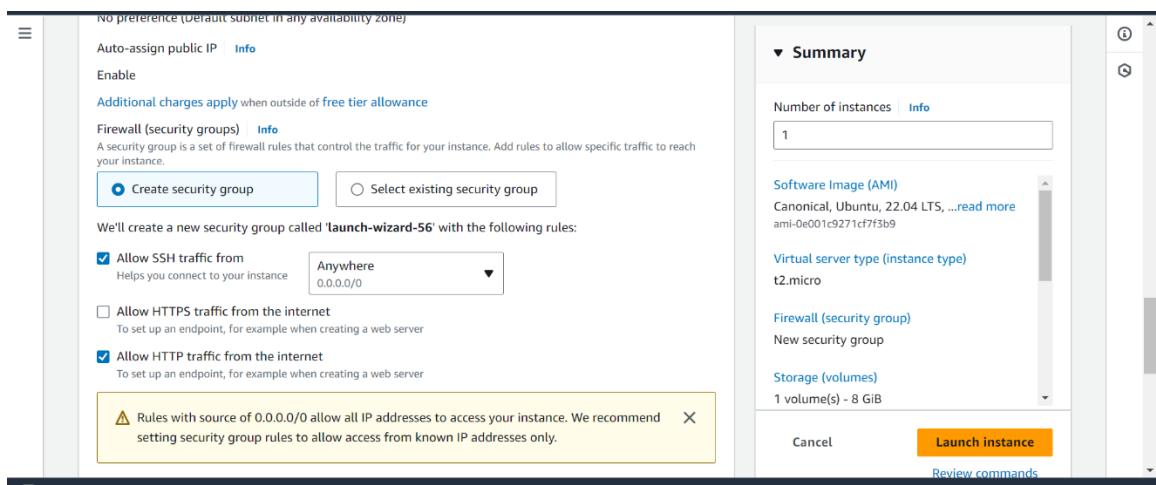
- Give tag name and key name as “ubuntu-docker” in "us-east-1".



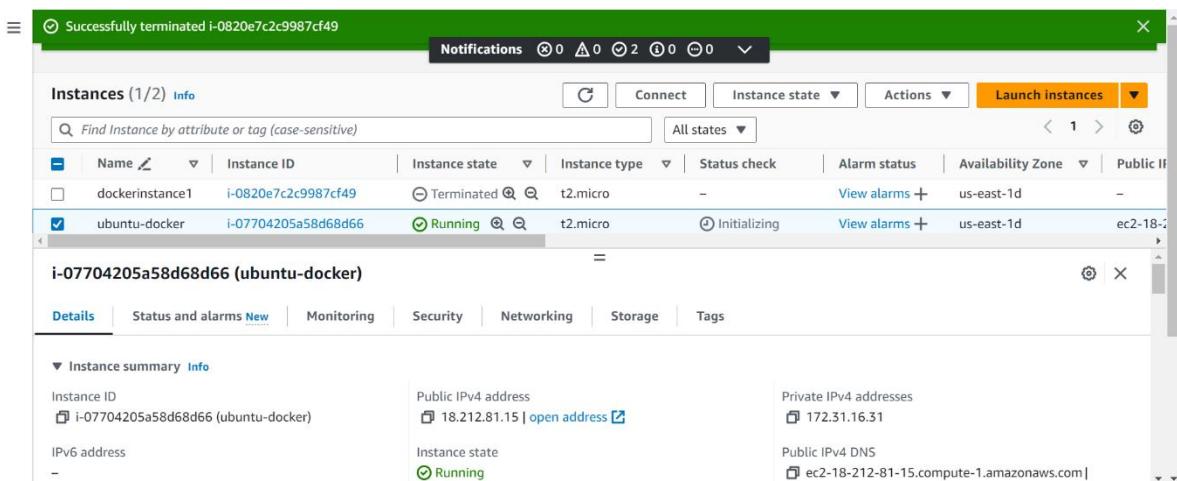
- EC2 instance AMI should be “ubuntu”.



- Allow SSH for SSH client for instance connection.



- Allow HTTP traffic from the internet for reaching website requests.



Write the following Steps :

```
sudo su -
ls
curl -fsSL https://get.docker.com -o get-docker.sh
ls
sh get-docker.sh
docker --version
docker pull httpd
docker run --name Apache-server -p 8080:80 httpd
```

```
logout
ubuntu@ip-172-31-16-31:~$ ls
ubuntu@ip-172-31-16-31:~$ sudo su -
root@ip-172-31-16-31:~# ls
get-docker.sh snap
root@ip-172-31-16-31:~# sh get-docker.sh
# Executing docker install script, commit: e5543d473431b782227f8908005543bb4389b8de
+ sh -c apt-get update -qq >/dev/null
+ sh -c DEBIAN_FRONTEND=noninteractive apt-get install -y -qq apt-transport-https ca-certificates curl >/dev/null
+ sh -c install -m 0755 -d /etc/apt/keyrings
+ sh -c curl -fsSL "https://download.docker.com/linux/ubuntu/gpg" | gpg --dearmor --yes -o /etc/apt/keyrings/docker.gpg
+ sh -c chmod a+r /etc/apt/keyrings/docker.gpg
+ sh -c echo "deb [arch=amd64 signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu jammy stable" > /etc/apt/sources.list.d/docker.list
+ sh -c apt-get update -qq >/dev/null
+ sh -c DEBIAN_FRONTEND=noninteractive apt-get install -y -qq docker-ce docker-ce-cli containerd.io docker-compose-plugin docker-ce-rootless-extras docker-buildx-plugin >/dev/null
+ sh -c docker version
Client: Docker Engine - Community
Version: 26.1.0

i-07704205a58d68d66 (ubuntu-docker)
PublicIPs: 18.212.81.15 PrivateIPs: 172.31.16.31
```

```
documentation for details: https://docs.docker.com/go/attack-surface/
=====
cot@ip-172-31-16-31:~# docker --version
docker version 26.1.0, build 9714adc
cot@ip-172-31-16-31:~# docker pull apache
sing default tag: latest
rror response from daemon: pull access denied for apache, repository does not exist or may require 'docker login': denied: requested ac
o the resource is denied
cot@ip-172-31-16-31:~# docker pull httpd
sing default tag: latest
atest: Pulling from library/httpd
0a0cf830b12: Pull complete
51c52adaa9b: Pull complete
f4fb700ef54: Pull complete
9d9f60535a6: Pull complete
43a2b3cf551: Extracting [=====] 9.961MB/26.03MB
a83e81966d6: Download complete
```

i-07704205a58d68d66 (ubuntu-docker)

```
AWS Services Search [Alt+S] N. Virginia Karthik.M
To run Docker as a non-privileged user, consider setting up the
Docker daemon in rootless mode for your user:
dockerd-rootless-setuptool.sh install
Visit https://docs.docker.com/go/rootless/ to learn about rootless mode.

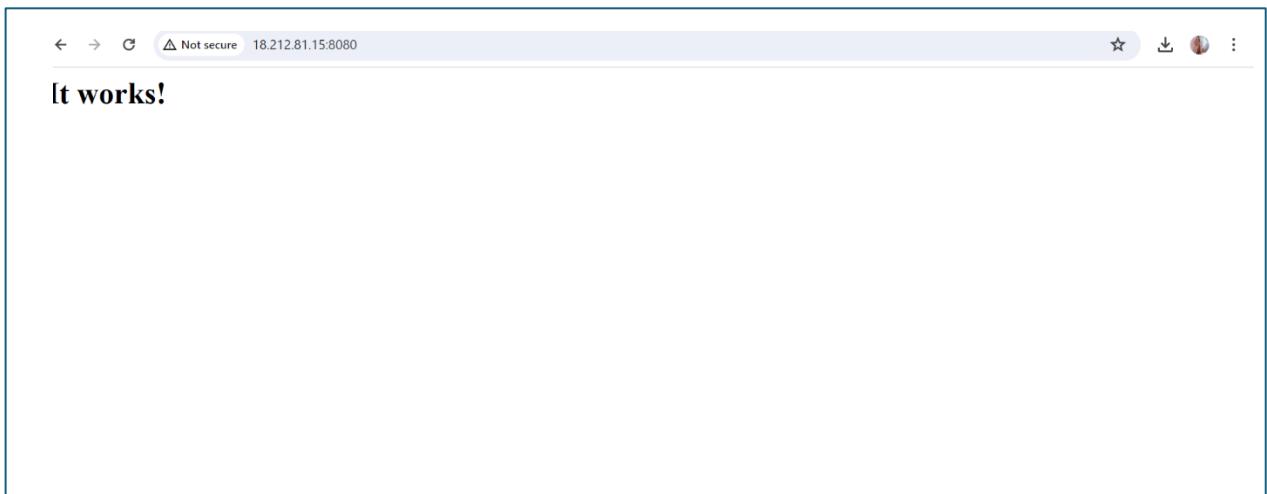
To run the Docker daemon as a fully privileged service, but granting non-root
users access, refer to https://docs.docker.com/go/daemon-access/
WARNING: Access to the remote API on a privileged Docker daemon is equivalent
to root access on the host. Refer to the 'Docker daemon attack surface'
documentation for details: https://docs.docker.com/go/attack-surface/
=====
root@ip-172-31-16-31:~# docker --version
Docker version 26.1.0, build 9714adc
root@ip-172-31-16-31:~#
```

i-07704205a58d68d66 (ubuntu-docker)

PublicIPs: 18.212.81.15 PrivateIPs: 172.31.16.31

- Install docker.
  - Pull docker Apache image from docker hub.
- Docker container name should be “Apache-server” and run docker in 8080 port, forward it to 80

Copy the Ip and paste it in the browser -><public-ip>:8080



**Note: The EC2 Instance type must be either t2.micro or t3.micro**

**RESULT:**



**SRI KRISHNA COLLEGE OF TECHNOLOGY  
(AN AUTONOMOUS INSTITUTION)**

**KOVAIPUDUR, COIMBATORE 641042**



<b>EXPERIMENT NO.</b>	:
<b>TITLE OF EXPERIMENT</b>	:
<b>DATE OF EXPERIMENT</b>	:

**EVALUATION BY FACULTY MEMBER (BASED ON RUBRICS)**

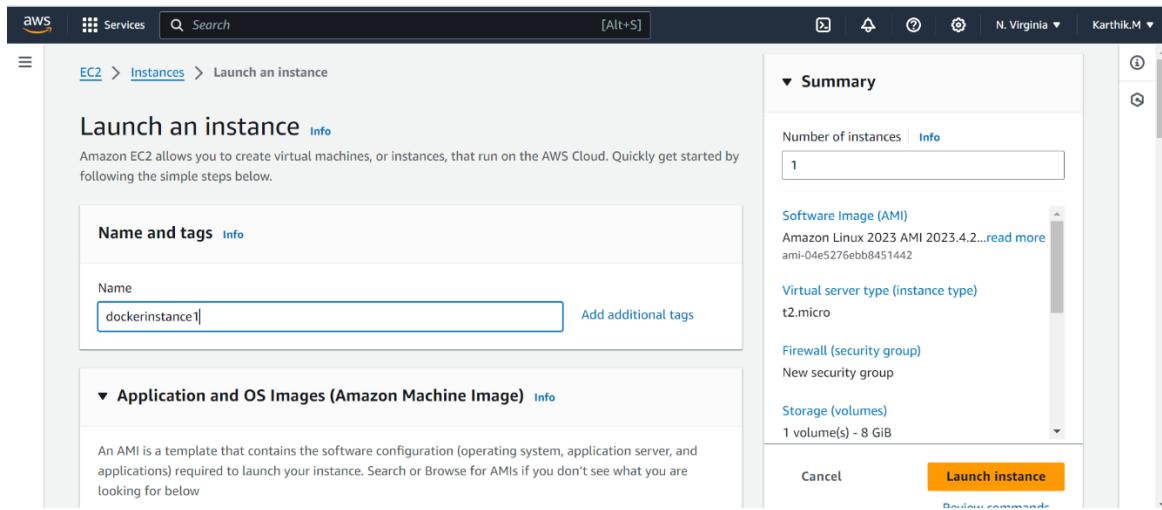
<b>CRITERIA</b>	<b>MAXIMUM MARKS</b>	<b>MARKS SCORED BY STUDENT</b>
Aim, Algorithm, Flowchart (optional)	20	
Coding and Commands	30	
Compilation and Debugging	30	
Execution and Result	10	
Documentation -Explanation of the experiment/ Viva	10	
<b>TOTAL MARKS</b>	<b>100</b>	
Signature of the faculty		

7. Create a public ECR with the name 'docker' in the region 'us-east-1' and push the docker image to the repository

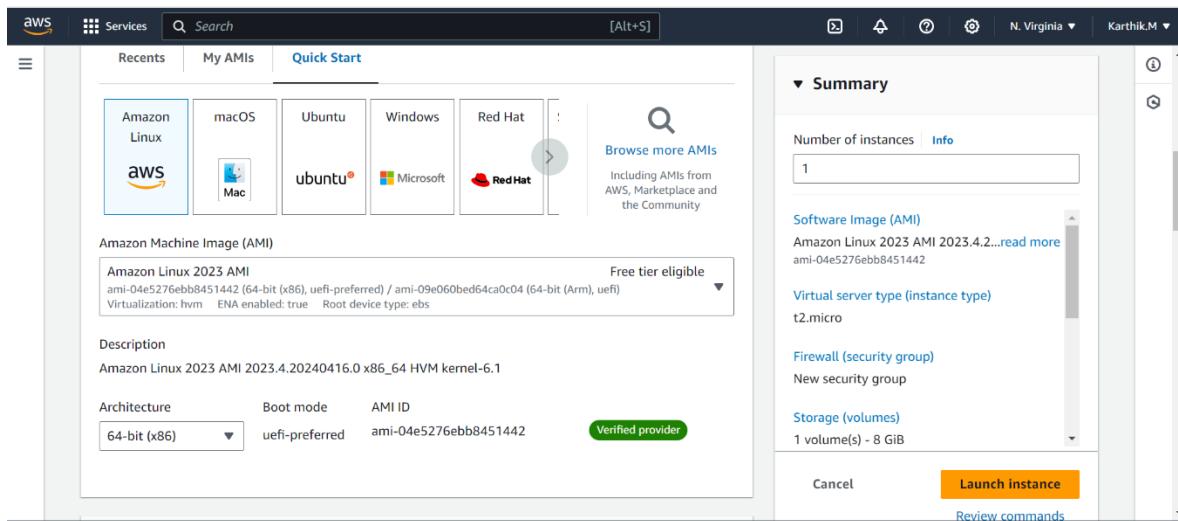
**AIM:**

## ALOGRITHM:

- Create an EC2 instance with the name tag "dockerinstance1" in the "us-east-1" region



- EC2 instance AMI should be "Amazon Linux 2023".



- Allow HTTP traffic from the instance security group

The screenshot shows the AWS Security Groups console for a new instance named 'Karthik.M'. It displays two security group rules:

- Security group rule 1 (TCP, 22, 0.0.0.0/0)**:
  - Type: ssh
  - Protocol: TCP
  - Port range: 22
  - Source type: Anywhere
  - Description: e.g. SSH for admin desktop
- Security group rule 2 (All, All, 0.0.0.0/0)**:
  - Type: All traffic
  - Protocol: All
  - Port range: All
  - Source type: Anywhere
  - Description: e.g. SSH for admin desktop

On the right side, the 'Summary' section shows 1 instance, the software image is Amazon Linux 2023.4.2, and the virtual server type is t2.micro. A 'Launch instance' button is visible.

- Install docker in the "dockerinstance1"

The screenshot shows the AWS EC2 Instances console. On the left, the navigation menu is expanded to show 'Instances' and its sub-options like 'Instance Types', 'Launch Templates', etc. The main table lists one instance:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Ava
dockerinstance1	i-0820e7c2c9987cf49	Pending	t2.micro	-	View alarms +	us-e

A modal window titled 'Select an instance' is open over the table.

- Write a dockerfile with httpd as its base image and build the file
- Install docker in the "dockerinstance1"

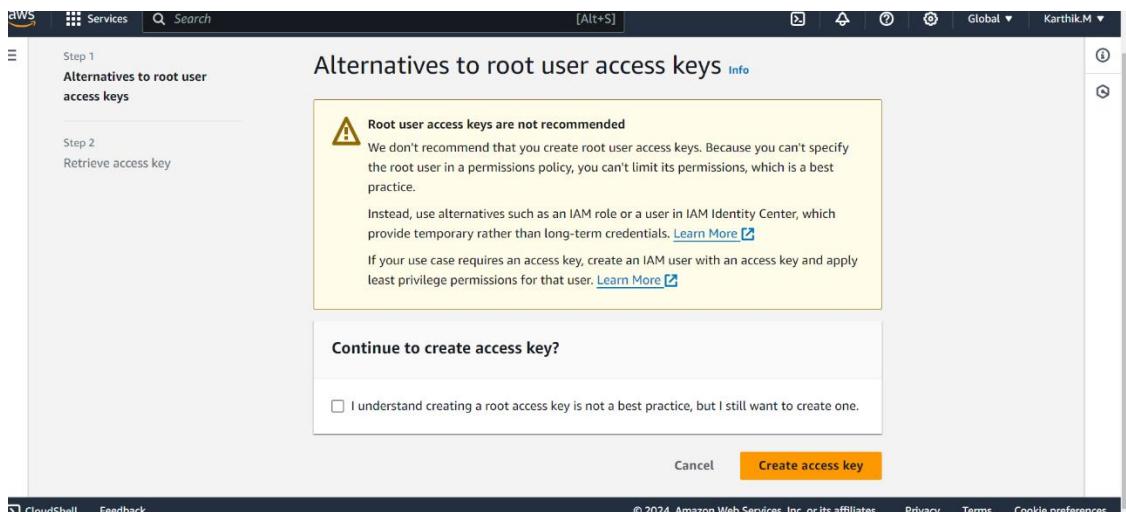
```
sudo su
yum update -y
sudo yum install -y docker
sudo service docker start
sudo usermod -a -G docker ec2-user
docker version

vim Dockerfile
```

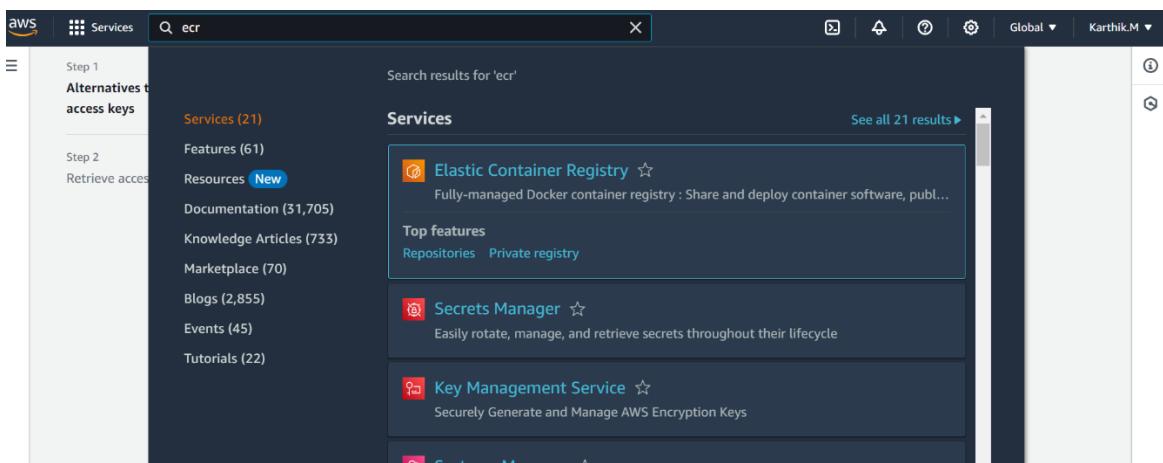
```
FROM httpd:2.4
# Copy your website files (replace with your directory)
COPY website /var/www/html/
# Expose the default Apache port (80)
EXPOSE 80
# Run httpd in the foreground
CMD ["httpd", "-D", "FOREGROUND"]
```

i-0820e7c2c9987cf49 (dockerinstance1)  
PublicIPs: 3.85.123.173 PrivateIPs: 172.31.16.164

```
vi index.html  
docker build -t latest .  
docker images  
aws configure  
Access key : .....  
secret key: .....
```



- Create public repository in ECR with the name "docker-repo1"



- Push the image to the created repository

The screenshot shows the ECR console with the URL <https://us-east-1.console.aws.amazon.com/ecr/public-registry/repositories?region=us-east-1>. The left sidebar has sections for Private registry (Repositories, Settings), Public registry (Repositories, Settings), ECR public gallery (Amazon ECS, Amazon EKS), Getting started, Documentation, CloudShell, and Feedback. The main area is titled 'Public repositories' and shows a table with one row for 'docker-repo1'. The table has columns for Repository name, URI, and Created at. The 'Create repository' button is highlighted in orange.

- The image should have 'latest' as its tag

The screenshot shows a modal dialog titled 'Push commands for docker-repo1'. It contains instructions for pushing an image to the repository. Step 1: Retrieve an authentication token and authenticate your Docker client to your registry. Use the AWS CLI: 

```
aws ecr-public get-login-password --region us-east-1 | docker login --username AWS --password-stdin public.ecr.aws/o7t6l3i3
```

. Note: if you receive an error using the AWS CLI, make sure that you have the latest version of the AWS CLI and Docker installed. Step 2: Build your Docker image using the following command. For information on building a Docker file from scratch, see the instructions [here](#). You can skip this step if your image has already been built: 

```
docker build -t docker-repo1 .
```

. Step 3: After the build is completed, tag your image so you can push the image to this repository: 

```
docker tag docker-repo1:latest public.ecr.aws/o7t6l3i3/docker-repo1:latest
```

. Step 4: Run the following command to push this image to your newly created AWS repository: 

```
aws ecr-public put-image --repository-name docker-repo1 --image-tag latest --image-digest sha256:f253a3fce933835c7cfcd5b99e641c741d6faecdca2a6e01444b534d26494ee
```

.

```

>> [internal] load build context
=> transferring context: 90B
=> [1/2] FROM docker.io/library/httpd:latest@sha256:36c8c79f900108f0f09fd4148ad35ade57cba0dc19d13f3d15be24ce94e6a639
=> CACHED [2/2] COPY index.html /usr/local/apache2/htdocs/
=> exporting to image
=> exporting layers
=> => writing image sha256:9709cc8782900fd407d650207c1b29d4a9320467d975c97e8b61b59e03994dce
=> => naming to docker.io/library/docker-repo1
[root@ip-172-31-16-164 ec2-user]# docker tag docker-repo1:latest public.ecr.aws/o7t6l3i3/docker-repo1:latest
[root@ip-172-31-16-164 ec2-user]# docker push public.ecr.aws/o7t6l3i3/docker-repo1:latest
The push refers to repository [public.ecr.aws/o7t6l3i3/docker-repo1]
87fbcc10c39d: Pushed
dball169a4efb: Pushed
edb961e754f: Pushed
53920c8b9c11: Pushed
5f70bf18a086: Pushed
46176a0cbe9f: Pushed
52ec5a4316fa: Pushed
latest: digest: sha256:f253a3fce933835c7cfcd5b99e641c741d6faecdca2a6e01444b534d26494ee size: 1779
[root@ip-172-31-16-164 ec2-user]# i-0820e7c2c9987cf49 (dockerinstance1)
PublicIPs: 3.85.123.173 PrivateIPs: 172.31.16.164

```

- The image should have 'latest' as its tag

The screenshot shows the AWS ECR console. The top navigation bar includes the AWS logo, a search bar with 'ec2', and account information for 'N. Virginia' and 'Karthik.M'. The main title is 'Amazon Elastic Container Registry'. On the left, a sidebar shows 'Private registry' and 'Public registry' sections, with 'Images' selected under 'Public registry'. The main content area displays a repository named 'docker-repo1'. A banner at the top of this area states: 'Image scan overview, status, and full vulnerabilities are now displayed in the Image detail page. To access, click an image tag.' Below this, the repository name 'docker-repo1' is shown, along with three buttons: 'View public listing', 'View push commands', and 'Edit'. A table titled 'Images (1)' lists the single image available. The table columns are: Image tag, Artifact type, Pushed at, Size (MB), Image URI, and Digest. The data row shows: 'latest', 'Image', '28 April 2024, 16:40:29 (UTC+05:5)', '59.19', 'Copy URI', and 'sha256:f253a3fc...'. A search bar labeled 'Search artefacts' is also present.

**Note:** While creating any AWS resources as per question either in Terraform or console, if you see any error saying that the resource "already exists" then kindly delete the existing resource and create your new resource. If VPC limit exceeds also, then kindly delete the existing VPC and create your new VPC

**RESULT:**



**SRI KRISHNA COLLEGE OF TECHNOLOGY  
(AN AUTONOMOUS INSTITUTION)**

**KOVAIPUDUR, COIMBATORE 641042**



<b>EXPERIMENT NO.</b>	:
<b>TITLE OF EXPERIMENT</b>	:
<b>DATE OF EXPERIMENT</b>	:

**EVALUATION BY FACULTY MEMBER (BASED ON RUBRICS)**

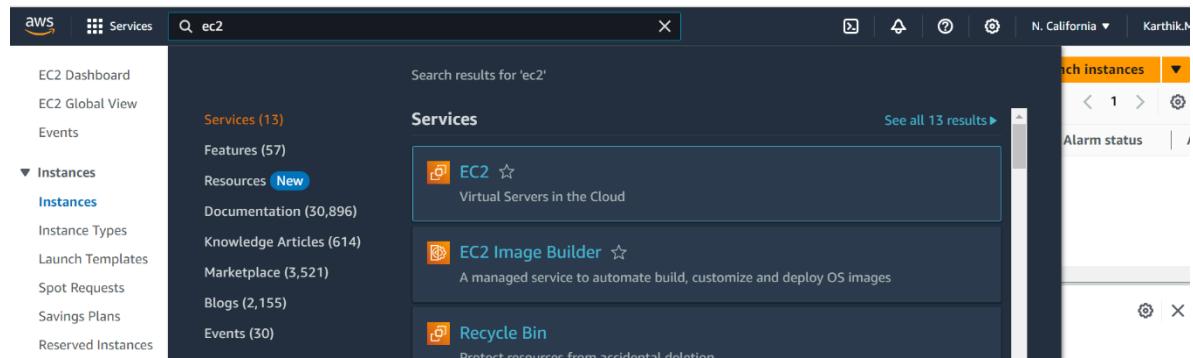
<b>CRITERIA</b>	<b>MAXIMUM MARKS</b>	<b>MARKS SCORED BY STUDENT</b>
Aim, Algorithm, Flowchart (optional)	20	
Coding and Commands	30	
Compilation and Debugging	30	
Execution and Result	10	
Documentation -Explanation of the experiment/ Viva	10	
<b>TOTAL MARKS</b>	<b>100</b>	
Signature of the faculty		

## 8. Configure an AWS Application Load Balancer to distribute traffic to EC2 instances across multiple availability zones within a single region.

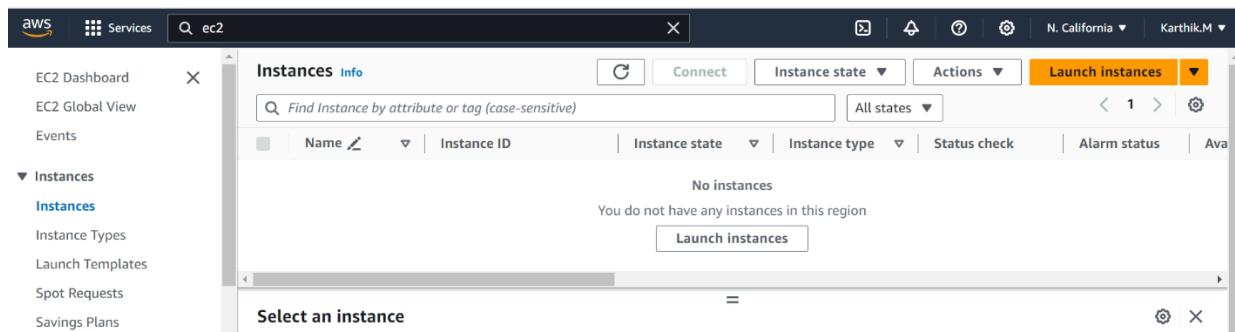
**AIM:**

**ALGORITHM:**

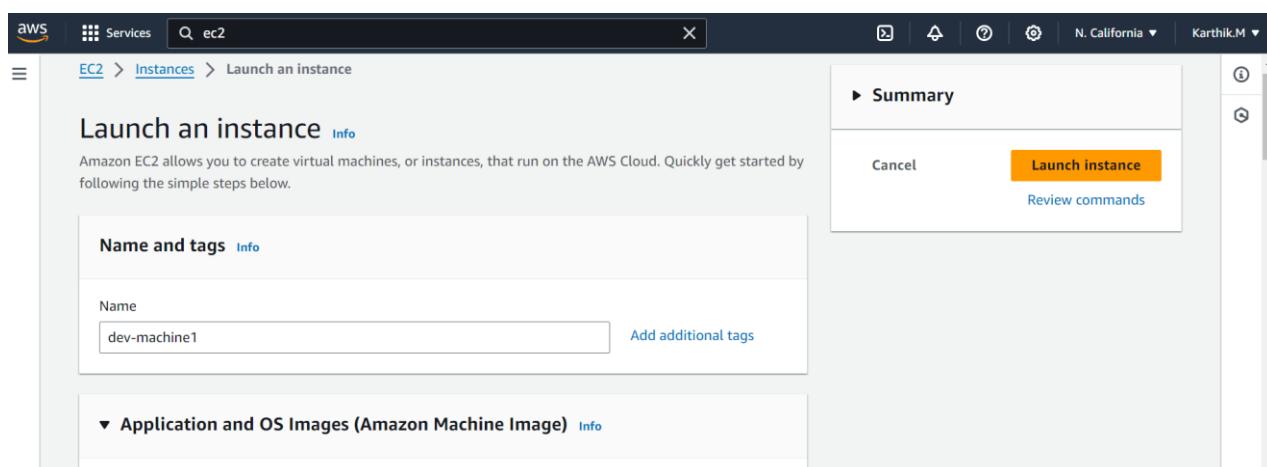
- Create an EC2 instance with the name tag "dev-machine1" in the "us-west-1b" availability zone



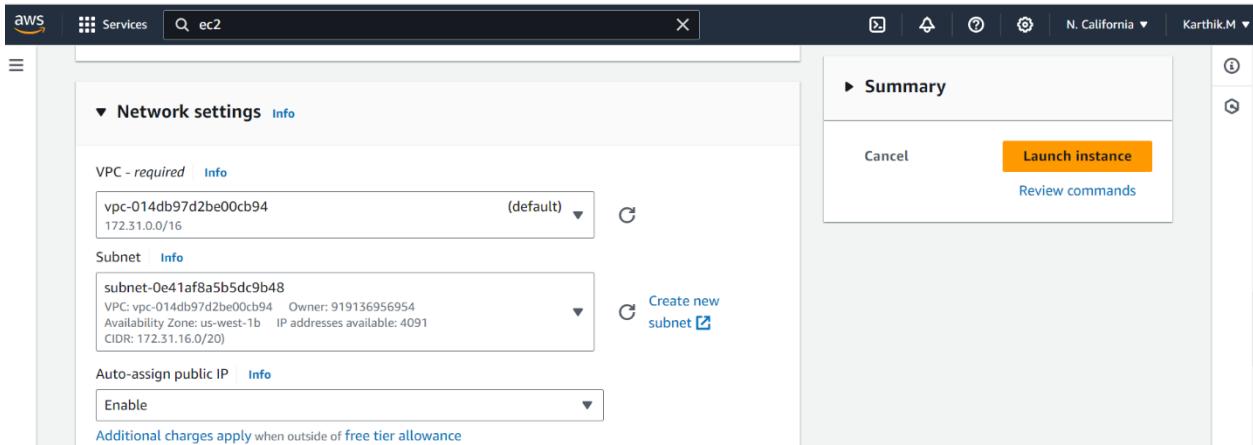
- Click launch instance



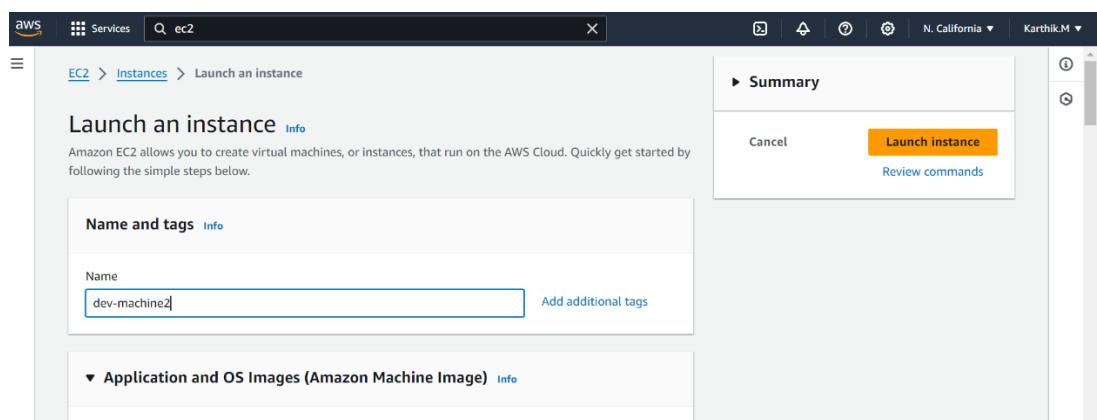
- Enter the name for Ec2 machine



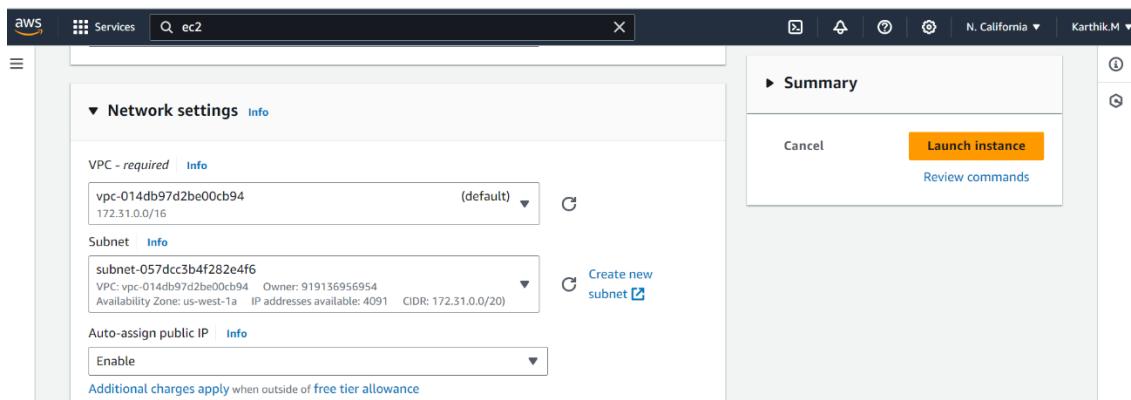
- Click edit -> Choose Availability zone us-west-1b



- Create another EC2 instance with the name tag "dev-machine2" in the "us-west-1c" availability zone
- Enter the name for second Ec2 machine



- Choose Availability Zone us-west-1c



- Configure a http website in both the instances and allow http traffic (Port: 80) in the security group

Go to Security groups, Click on Add security group

The screenshot shows the 'Create security group' step in the EC2 Security Groups wizard. It includes fields for 'Security group name - required' (set to 'launch-wizard-24'), 'Description - required' (set to 'launch-wizard-24 created 2024-04-28T04:29:22.809Z'), and an 'Inbound Security Group Rules' section which is currently empty. On the right, a summary panel shows 'Launch instance' as the next step.

- Enable Http traffic (port:80) in both the instance.

The screenshot shows the 'Inbound Security Group Rules' configuration. It contains two rules: one for SSH (TCP port 22) allowing traffic from anywhere, and one for HTTP (TCP port 80) also allowing traffic from anywhere. Both rules have optional descriptions. The right side of the screen shows a summary panel with 'Launch instance' highlighted.

- Create a Target group with the name "dev-tg"

Go to Target groups, Click on create Target group

The screenshot shows the 'Target groups' page. The left sidebar has sections for Volumes, Snapshots, Lifecycle Manager, Network & Security (Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces), Load Balancing (Load Balancers, Target Groups, Trust Stores), and Auto Scaling (Auto Scaling Groups). The 'Target Groups' section is currently selected. The main area displays a table with columns for Name, ARN, Port, Protocol, and Target type, showing 'No target groups'. A large callout at the bottom says '0 target groups selected' and 'Select a target group above.' with a 'Create target group' button.

- Choose instance.

**Specify group details**

Your load balancer routes requests to the targets in a target group and performs health checks on the targets.

**Basic configuration**

Settings in this section can't be changed after the target group is created.

**Choose a target type**

Instances

- Supports load balancing to instances within a specific VPC.
- Facilitates the use of [Amazon EC2 Auto Scaling](#) to manage and scale your EC2 capacity.

IP addresses

- Supports load balancing to VPC and on-premises resources.
- Facilitates routing to multiple IP addresses and network interfaces on the same instance.
- Offers flexibility with microservice based architectures, simplifying inter-application communication.
- Supports IPv6 targets, enabling end-to-end IPv6 communication, and IPv4-to-IPv6 NAT.

- Enter the name for target group-> click next.

**Target group name**

dev-tg

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

**Protocol : Port**

Choose a protocol for your target group that corresponds to the Load Balancer type that will route traffic to it. Some protocols now include anomaly detection for the targets and you can set mitigation options once your target group is created. This choice cannot be changed after creation

TCP 80  
1-65535

**IP address type**

Only targets with the indicated IP address type can be registered to this target group.

IPv4

Each instance has a default network interface (eth0) that is assigned the primary private IPv4 address. The instance's primary private IPv4 address is the one that will be applied to the target.

IPv6

Each instance you register must have an assigned primary IPv6 address. This is configured on the instance's default network interface (eth0). [Learn more](#)

- Register the above created two instances in the "dev-tg" Target Group  
Choose both the instance and give "include as pending below"

**Available instances (2/2)**

Instance ID	Name	State	Security group
i-007f7f5aaa8a7c454	dev-machine2	Running	launch-wizard
i-07b82b630cae0eb35	dev-machine1	Running	launch-wizard

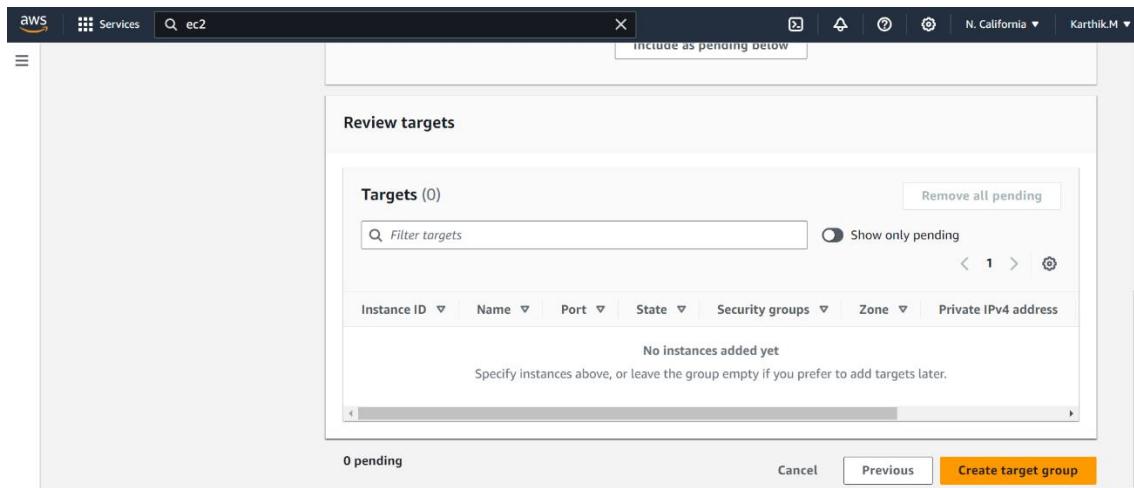
2 selected

**Ports for the selected instances**

Ports for routing traffic to the selected instances.

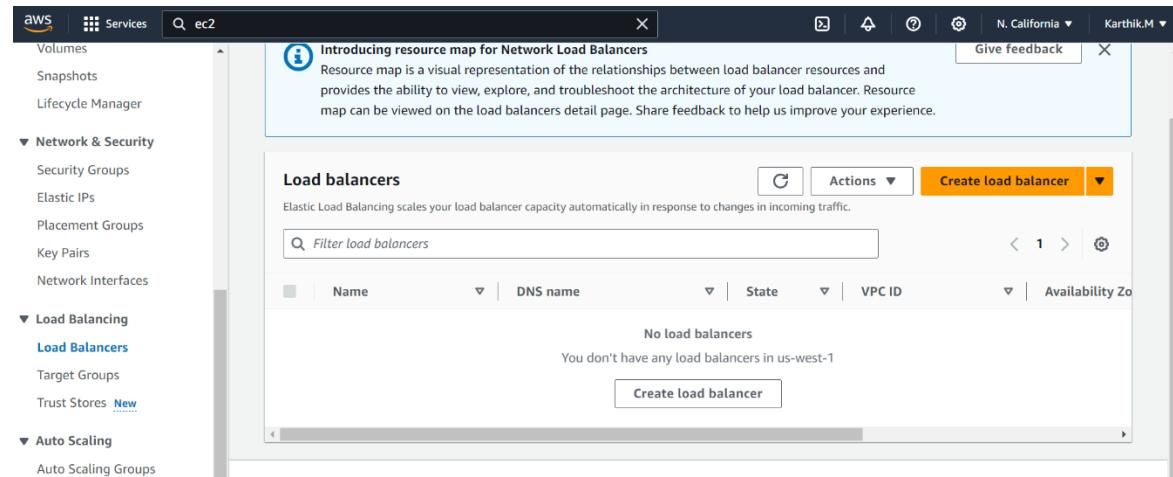
80  
1-65535 (separate multiple ports with commas)

**Include as pending below**

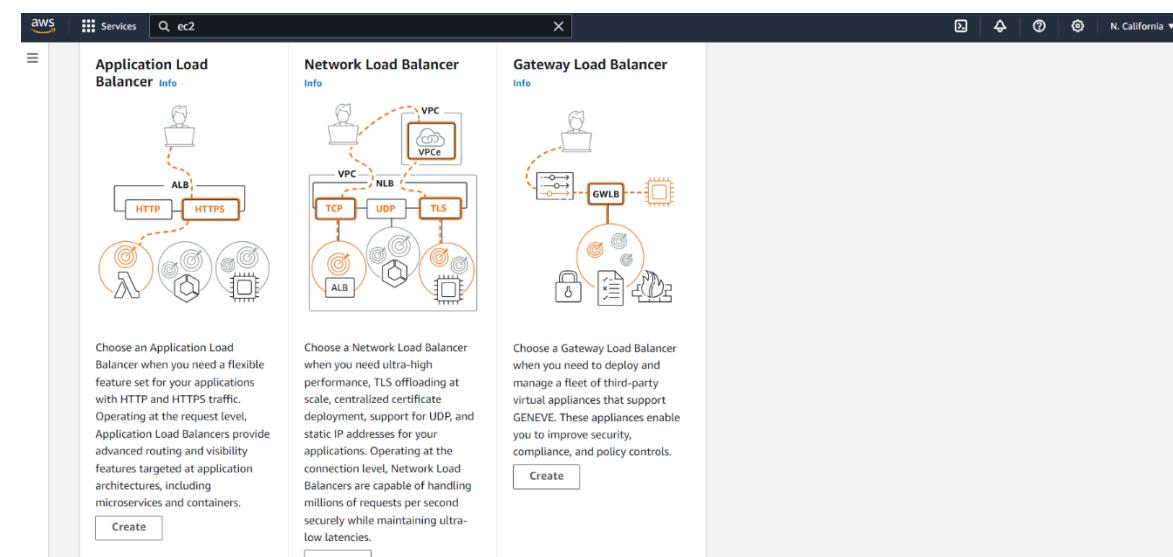


- Click Create Target group
- Create Loadbalancer with the name "dev-lb" by using "dev-tg" Target Group

Go to load balancer, Click on create load balancer



- Create Application load balancer



- Enter the name

The screenshot shows the 'Create Application Load Balancer' page in the AWS EC2 console. In the 'Basic configuration' section, the 'Load balancer name' field is set to 'dev-lb'. Below it, the 'Scheme' dropdown is set to 'Info'. A note states that the scheme cannot be changed after creation.

- Select Target groups

The screenshot shows the 'Listeners and routing' configuration for the load balancer. Under the 'Listener HTTP:80' section, the 'Protocol' is set to 'HTTP' and the 'Port' is '80'. The 'Default action' dropdown is set to 'Select a target group', and a dropdown menu shows 'dev-tg' selected. The 'Listener tags - optional' section includes a 'Add listener tag' button.

- Click on Create

The screenshot shows the final step of creating the load balancer. It displays the 'Creation workflow and status' section with a note about server-side tasks and their statuses. At the bottom right, there are 'Cancel' and 'Create load balancer' buttons.

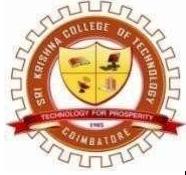
**Note:** While creating any AWS resources as per question in console, if you see any error saying that the resource "already exists" then kindly delete the existing resource and create your new resource. If VPC limit exceeds also, then kindly delete the existing VPC and create your new VPC

**RESULT:**



**SRI KRISHNA COLLEGE OF TECHNOLOGY  
(AN AUTONOMOUS INSTITUTION)**

**KOVAIPUDUR, COIMBATORE 641042**



<b>EXPERIMENT NO.</b>	:
<b>TITLE OF EXPERIMENT</b>	:
<b>DATE OF EXPERIMENT</b>	:

**EVALUATION BY FACULTY MEMBER (BASED ON RUBRICS)**

<b>CRITERIA</b>	<b>MAXIMUM MARKS</b>	<b>MARKS SCORED BY STUDENT</b>
Aim, Algorithm, Flowchart (optional)	20	
Coding and Commands	30	
Compilation and Debugging	30	
Execution and Result	10	
Documentation -Explanation of the experiment/ Viva	10	
<b>TOTAL MARKS</b>	<b>100</b>	
Signature of the faculty		

## 9. Create a docker container with the following requirements.

**AIM:**

**ALGORITHM:**

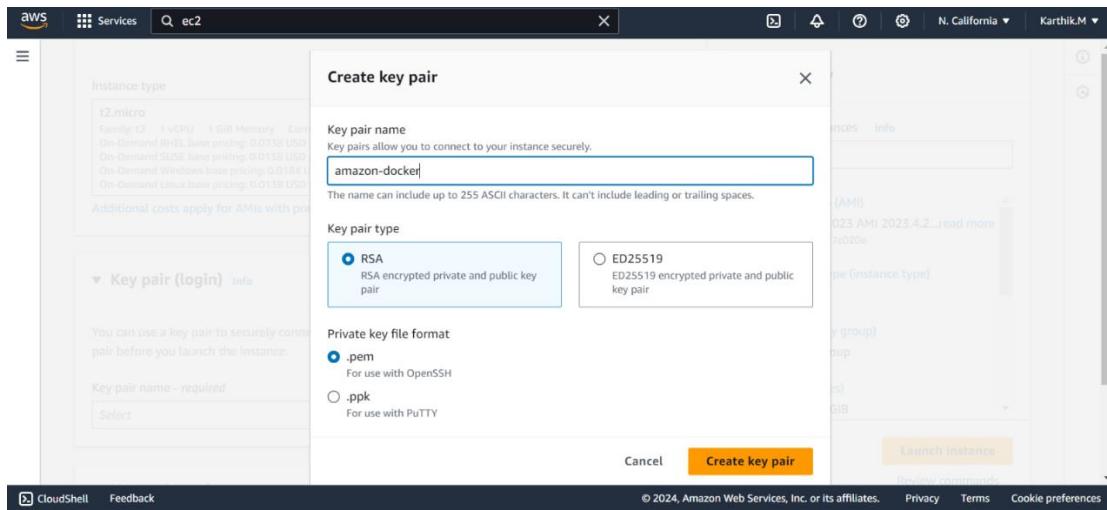
- **Requirements:**
- Tag name and key name: "amazon-docker".  
Launch Ec2-instance and enter the name

The screenshot shows the 'Launch an instance' wizard in the AWS Management Console. The 'Name and tags' section has 'Name' set to 'amazon-docker'. The 'Software Image (AMI)' section shows 'Amazon Linux 2023 AMI 2023.4.2...' with AMI ID 'ami-0827b6c5b977c020e'. The 'Virtual server type (instance type)' is set to 't2.micro'. The 'Launch instance' button is highlighted in orange at the bottom right.

- EC2 instance AMI: Amazon Linux 2023.

The screenshot shows the 'Quick Start' page for selecting an AMI. It features a grid of icons for different operating systems: Amazon Linux, Ubuntu, Windows, Red Hat, and SUSE Linux. Below the grid, the 'Amazon Machine Image (AMI)' section displays 'Amazon Linux 2023 AMI' with AMI ID 'ami-0827b6c5b977c020e'. The 'Description' section notes it is 'Free tier eligible' and provides technical details like 'Architecture: 64-bit (x86)', 'Boot mode: uefi-preferred', and 'AMI ID: ami-0827b6c5b977c020e'. The 'Verified provider' badge is present. To the right, the 'Summary' section mirrors the configuration from the previous screenshot, including the same AMI selection and 'Launch instance' button.

- Key pair name : "amazon-docker"



- Allow SSH for SSH client for instance connection.
- Allow HTTPS traffic from the internet for secure connections.

- Install Docker.
- Build Docker image for Apache2 server with index.html using Dockerfile.
- Run Docker container from the built image

```
sudo su
yum update -y
sudo yum install -y docker
sudo service docker start
sudo usermod -a -G docker ec2-user
docker version
docker ps
docker pull httpd
docker run -d -p 80:80 httpd
```

```
sudo: amazon-linux-extras: command not found
[root@ip-172-31-31-239 ~]# sudo yum install -y docker
Last metadata expiration check: 0:02:34 ago on Sun Apr 28 07:08:23 2024.
Dependencies resolved.

=====
| Package           | Architecture | Version      | Repository | Size   |
|=====|
| docker           | x86_64       | 25.0.3-1.amzn2023.0.1 | amazonlinux | 44 M  |
| Installing:      |             |              |            |        |
| docker           | x86_64       | 1.7.11-1.amzn2023.0.1 | amazonlinux | 35 M  |
| containerd       | x86_64       | 1.8.8-3.amzn2023.0.2 | amazonlinux | 401 k |
| iptables-libc    | x86_64       | 1.8.8-3.amzn2023.0.2 | amazonlinux | 183 k |
| iptables-nft     | x86_64       | 3.0-1.amzn2023.0.1  | amazonlinux | 75 k  |
| libcgroup        | x86_64       | 1.0.8-2.amzn2023.0.2 | amazonlinux | 58 k  |
| libnftfilter     | x86_64       | 1.0.1-19.amzn2023.0.2 | amazonlinux | 30 k  |
| libnftnl         | x86_64       | 1.2.2-2.amzn2023.0.2 | amazonlinux | 84 k  |
| pigz             | x86_64       | 2.5-1.amzn2023.0.3  | amazonlinux | 83 k  |
| runc             | x86_64       | 1.1.11-1.amzn2023.0.1 | amazonlinux | 3.0 M  |

i-0d956ad520423be89 (amazon-docker)
PublicIPs: 54.86.57.51 PrivateIPs: 172.31.31.239
```

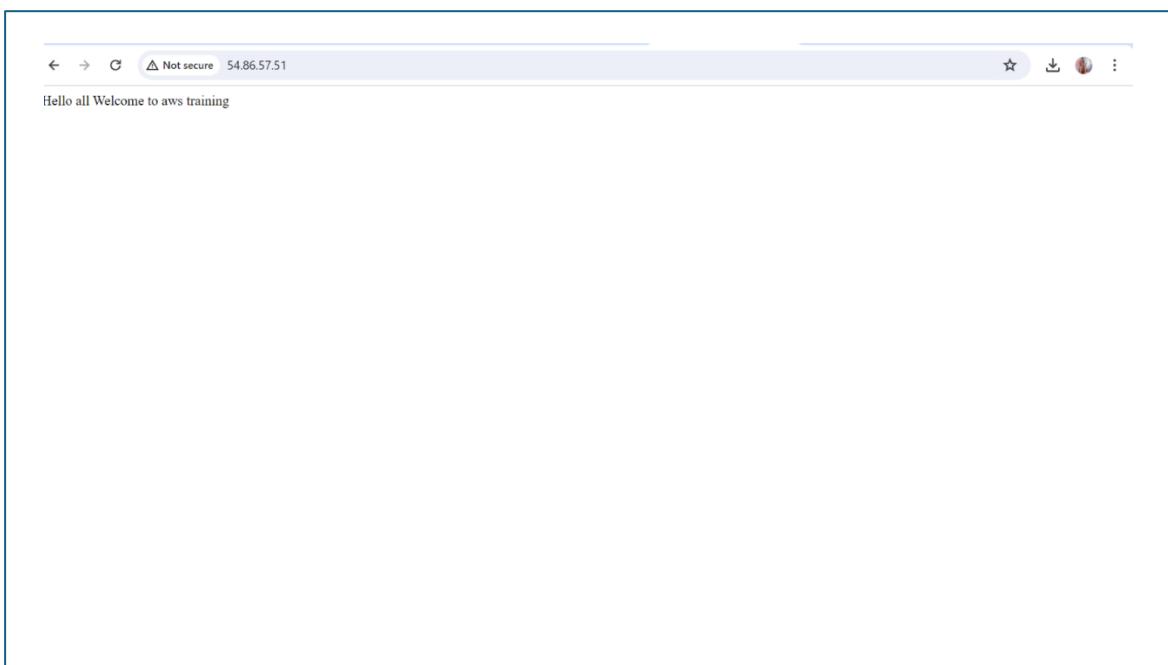


```
docker exec -it <container-id> bashcd /usr/local/apache2/htdocs/
```

```
ls
```

```
cat > index.html
```

- Copy the Ip and paste it in browser



- Note: The EC2 Instance type must be either t2.micro or t3.micro

**RESULT:**



**SRI KRISHNA COLLEGE OF TECHNOLOGY  
(AN AUTONOMOUS INSTITUTION)**

**KOVAIPUDUR, COIMBATORE 641042**



<b>EXPERIMENT NO.</b>	:
<b>TITLE OF EXPERIMENT</b>	:
<b>DATE OF EXPERIMENT</b>	:

**EVALUATION BY FACULTY MEMBER (BASED ON RUBRICS)**

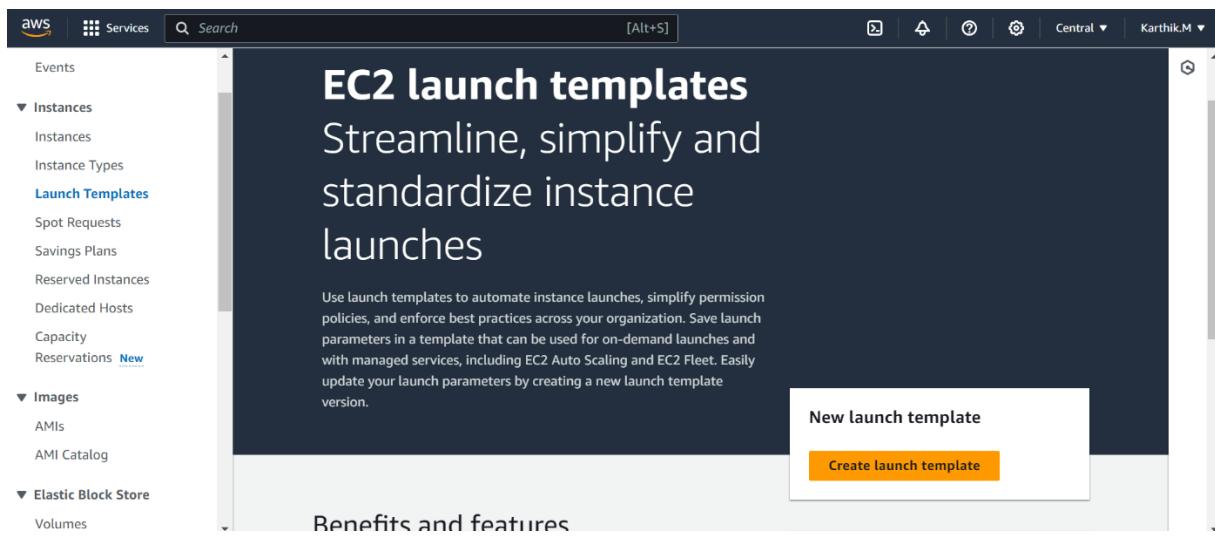
<b>CRITERIA</b>	<b>MAXIMUM MARKS</b>	<b>MARKS SCORED BY STUDENT</b>
Aim, Algorithm, Flowchart (optional)	20	
Coding and Commands	30	
Compilation and Debugging	30	
Execution and Result	10	
Documentation -Explanation of the experiment/ Viva	10	
<b>TOTAL MARKS</b>	<b>100</b>	
Signature of the faculty		

10. Create a launch template and auto-scaling group in the region "ca-central-1" with the below requirements

**AIM:**

**ALGORITHM:**

- Create a Launch Template in the name "sample1"
- Search Ec2, Go to Launch template



Choose Amazon linux 2023 version

- The instance AMI Should be "Amazon Linux 2023"
- The Instance Type should be "t2.micro" or "t3.micro"

The screenshot shows the AWS Launch Wizard interface. On the left, there's a sidebar with navigation icons. The main area displays the following details:

- Architecture:** 64-bit (x86)
- Boot mode:** uefi-preferred
- AMI ID:** ami-085c5194d6f95060c (Verified provider)

**Instance type:** t2.micro (Free tier eligible)  
Family: t2 - 1 vCPU - 1 GiB Memory - Current generation: true  
On-Demand RHEL base pricing: 0.0728 USD per Hour  
On-Demand Windows base pricing: 0.0174 USD per Hour  
On-Demand SUSE base pricing: 0.0128 USD per Hour  
On-Demand Linux base pricing: 0.0128 USD per Hour

**Additional costs apply for AMIs with pre-installed software**

**Key pair (login):** [Info] Advanced

**Summary:**

- Software Image (AMI):** Amazon Linux 2023 AMI 2023.4.2...read more  
ami-085c5194d6f95060c
- Virtual server type (instance type):** t2.micro
- Firewall (security group):** -
- Storage (volumes):** 1 volume(s) - 8 GiB

**Free tier:** In your first year includes 750 hours of t2.micro

- The Instance Keypair name should be "sample1"

The screenshot shows the AWS Launch Wizard interface. The configuration is identical to the previous one, with the t2.micro instance type selected and the 8 GiB storage volume.

**Summary:**

- Software Image (AMI):** Amazon Linux 2023 AMI 2023.4.2...read more  
ami-085c5194d6f95060c
- Virtual server type (instance type):** t2.micro
- Firewall (security group):** -
- Storage (volumes):** 1 volume(s) - 8 GiB

**Free tier:** In your first year includes 750 hours of t2.micro

- The Instance Root volume size should be 15GB

The screenshot shows the EBS Volume configuration screen. A volume named "Volume 1 (AMI Root) (Custom)" is being configured:

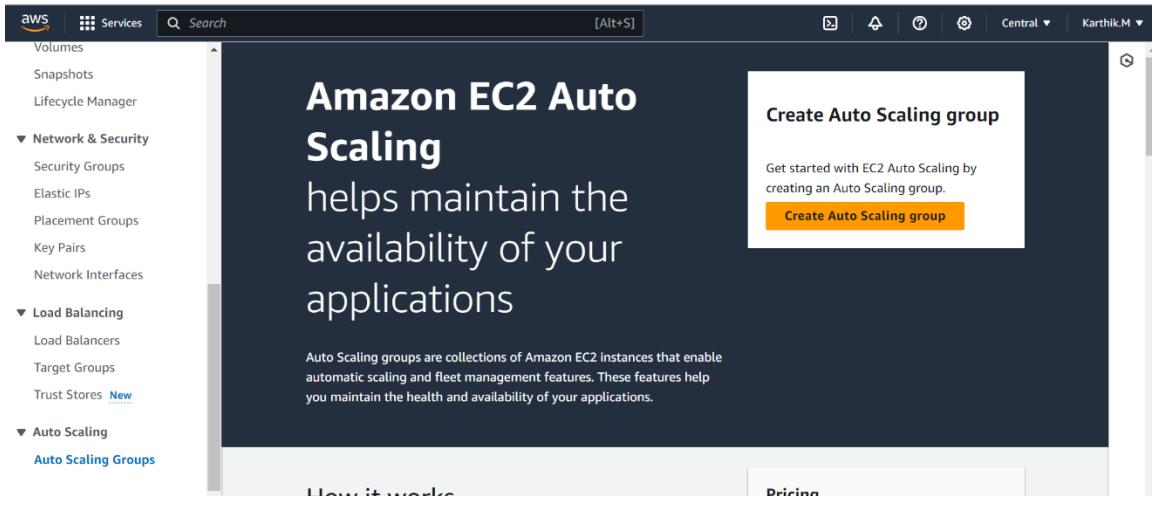
- Storage type:** EBS
- Device name:** /dev/xvda
- Snapshot:** snap-0c0a037fe212410c2
- Size (GiB):** 15
- Volume type:** gp3
- IOPS:** 3000
- Delete on termination:** Yes
- Encrypted:** Not encrypted
- KMS key:** Info
- Throughput:** 125

**Summary:**

- Software Image (AMI):** Amazon Linux 2023 AMI 2023.4.2...read more  
ami-085c5194d6f95060c
- Virtual server type (instance type):** t2.micro
- Firewall (security group):** -
- Storage (volumes):** 1 volume(s) - 15 GiB

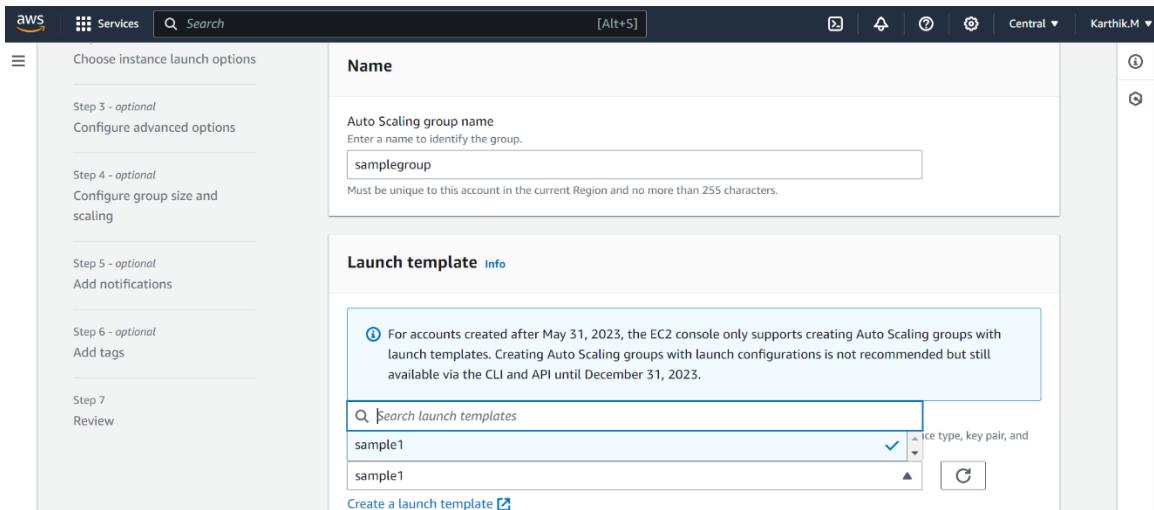
**Free tier:** In your first year includes 750 hours of t2.micro

- Create an auto-scaling group with the name "samplegroup"



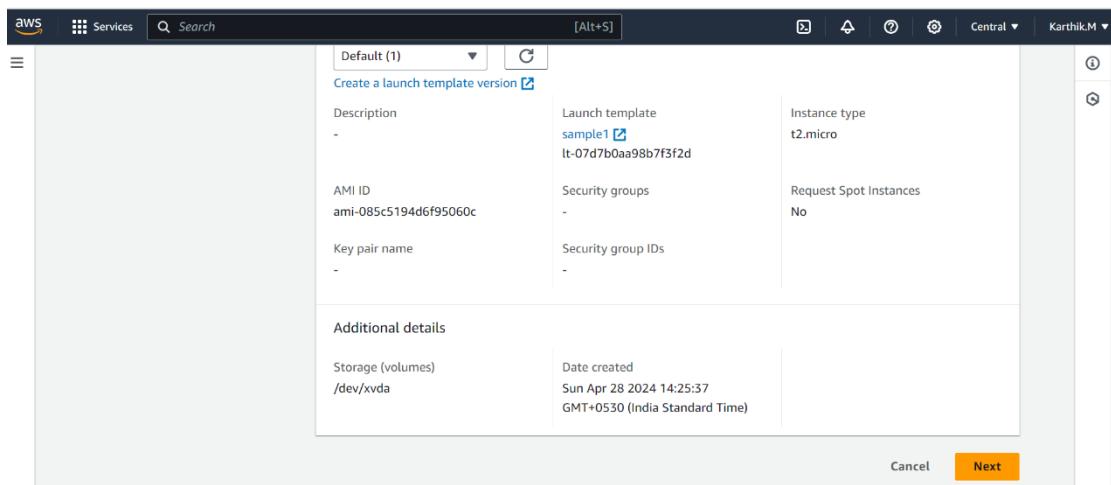
The screenshot shows the Amazon EC2 Auto Scaling landing page. On the left, there's a navigation sidebar with links like Volumes, Snapshots, Lifecycle Manager, Network & Security, Load Balancing, and Auto Scaling. Under Auto Scaling, 'Auto Scaling Groups' is selected. The main content area features a large heading 'Amazon EC2 Auto Scaling' with the subtext 'helps maintain the availability of your applications'. Below this, a paragraph explains that Auto Scaling groups are collections of Amazon EC2 instances that enable automatic scaling and fleet management features. A prominent orange button labeled 'Create Auto Scaling group' is visible.

- The group's availability zone should be ca-central-1a & ca-central-1b



This screenshot shows the third step of creating an Auto Scaling group: 'Choose instance launch options'. It includes optional steps for advanced options, group size, notifications, and tags. The 'Launch template' section shows a search bar with 'sample1' selected, and a note about the transition to launch templates. A 'Create a launch template' button is also present.

- The maximum capacity of the autoscaling group should be 4



This screenshot shows the fourth step of creating an Auto Scaling group: 'Configure advanced options'. It displays a table with configuration details: Default (1) launch template named 'sample1' (ID: lt-07d7b0aa98b7f3f2d), AMI ID 'ami-085c5194d6f95060c', Key pair name (empty), Security groups (empty), Request Spot Instances set to 'No', and storage information ('/dev/xvda'). Buttons for 'Cancel' and 'Next' are at the bottom.

**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**

Choose the VPC that defines the virtual network for your Auto Scaling group.

- ca-central-1a | subnet-02f8e6b0951690f21  
172.31.16.0/20 Default
- ca-central-1b | subnet-040f826866b339f47  
172.31.0.0/20 Default
- ca-central-1d | subnet-0621df1650393ad98  
172.31.32.0/20 Default

Select Availability Zones and subnets

- ca-central-1a | subnet-02f8e6b0951690f21 X  
172.31.16.0/20 Default
- ca-central-1b | subnet-040f826866b339f47 X  
172.31.0.0/20 Default

- The minimum capacity of the autoscaling group should be 2
- The desired capacity of the autoscaling group should be 2

**Configure advanced options - optional**

Integrate your Auto Scaling group with other services to distribute network traffic across multiple servers using a load balancer or to establish service-to-service communications using VPC Lattice. You can also set options that give you more control over health check replacements and monitoring.

**Load balancing**

Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.

- No load balancer  
Traffic to your Auto Scaling group will not be fronted by a load balancer.
- Attach to an existing load balancer  
Choose from your existing load balancers.
- Attach to a new load balancer  
Quickly create a basic load balancer to attach to your Auto Scaling group.

**VPC Lattice integration options**

To improve networking capabilities and scalability, integrate your Auto Scaling group with VPC Lattice. VPC Lattice facilitates communications between AWS services and helps you connect and manage your applications across compute services in AWS.

Enter the value

**Configure group size and scaling**

**Desired capacity type**

Choose the unit of measurement for the desired capacity value. vCPUs and Memory(GiB) are only supported for mixed instances groups configured with a set of instance attributes.

Units (number of instances)

**Desired capacity**

Specify your group size.

2

**Scaling**

You can resize your Auto Scaling group manually or automatically to meet changes in demand.

**Scaling limits**

Set limits on how much your desired capacity can be increased or decreased.

Min desired capacity	Max desired capacity
2	4

Equal or less than desired capacity      Equal or greater than desired capacity

**Automatic scaling - optional**

- Click ->next

The screenshot shows the 'Create Auto Scaling group' wizard at Step 5: 'Add notifications - optional'. The page title is 'Auto Scaling groups > Create Auto Scaling group'. On the left, there's a vertical navigation bar with steps: Step 1 (Choose launch template), Step 2 (Choose instance launch options), Step 3 (optional: Configure advanced options), Step 4 (optional: Configure group size and scaling), Step 5 (optional: Add notifications), and Step 6 (optional: Add tags). The main content area has a heading 'Add notifications - optional' with a link to 'Info'. It says 'Send notifications to SNS topics whenever Amazon EC2 Auto Scaling launches or terminates the EC2 instances in your Auto Scaling group.' Below this is a button labeled 'Add notification'. At the bottom right are buttons for 'Cancel', 'Skip to review', 'Previous', and 'Next' (highlighted in orange).

- Click create auto Scalling Group.

The screenshot shows the 'Create Auto Scaling group' wizard at Step 6: 'Add tags'. The page title is 'Auto Scaling groups > Create Auto Scaling group'. The left sidebar shows 'Step 5: Add notifications' (with 'Edit' button) and 'Step 6: Add tags' (with 'Edit' button). The main content area has a section titled 'Notifications' with a message 'No notifications'. Below it is another section titled 'Tags (0)' with a table header 'Key' | 'Value' | 'Tag new instances'. The table body shows 'No tags'. At the bottom right are buttons for 'Cancel', 'Previous', and 'Create Auto Scaling group' (highlighted in orange).

- Final Result :

The screenshot shows the 'Auto Scaling groups' list page. The title is 'EC2 > Auto Scaling groups'. The top navigation bar includes 'Services', 'Search', '[Alt+S]', and user information 'Central' and 'Karthik.M'. There are buttons for 'Actions' and 'Create Auto Scaling group'. A search bar is present. The main table lists one Auto Scaling group:

Name	Launch template/configuration	Instances	Status	Desired capacity	Min
<a href="#">samplegroup</a>	<a href="#">sample1</a>   Version Default	0	Updating capacity...	2	2

At the bottom, it says '0 Auto Scaling groups selected'.

**RESULT:**