

# COS20019 – CLOUD COMPUTING ARCHITECTURE

Name: Nguyen Linh Dan

Student ID: 103488557

## ACA MODULE 9 CHALLENGE LAB

### CREATING A SCALABLE AND HIGHLY AVAILABLE ENVIRONMENT FOR THE CAFE

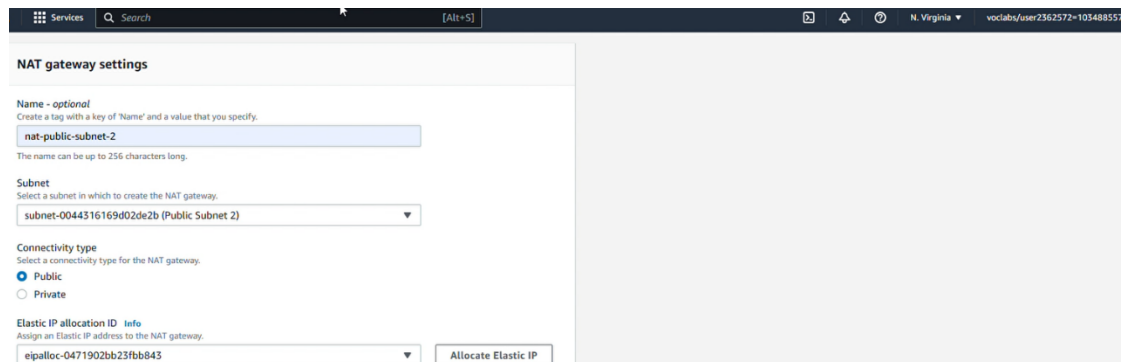
#### Task 1: Inspecting your environment

In this lab, I only go through the VPC settings to see how the VPC and subnets are configured. There is no configuration required in this part. I inspected that the initial infrastructure includes:

- 1 VPC with 2 Availability Zones
- 6 subnets (3 subnets/AZ)
- A security group called CafeSG is used for the following steps.
- The Route tables of all subnets were created, but they still need further configuration.

#### Task 2: Creating a NAT gateway for the second Availability Zone

First, I created a **NAT gateway** in the Public subnet 2. This NAT gateway would allow the scaling instances in the private subnets to access the Internet.



The screenshot shows the 'NAT gateway settings' page in the AWS Management Console. The 'Name - optional' field is set to 'nat-public-subnet-2'. The 'Subnet' dropdown is set to 'subnet-0044316169d02de2b (Public Subnet 2)'. The 'Connectivity type' is set to 'Public'. The 'Elastic IP allocation ID' is set to 'eipalloc-0471902bb23fb6843'. There is an 'Allocate Elastic IP' button.

After creating the NAT gateway, I need to create a **Route table** to add the NAT gateway and associate it with the private subnet 2. Luckily, the Private subnet 2 Route table was prepared by the unit convenor, so I only needed to edit it without creating it.



The screenshot shows the 'Edit routes' page in the AWS Management Console. The 'Destination' is set to '10.0.0.0/16'. The 'Target' dropdown is open, showing options: 'local', 'nat-0c9fe553805f48df (nat-public-subnet-2)', and 'nat-0347339adfe67124'. The 'Status' is 'Active' and 'Propagated' is 'No'. There is a 'Remove' button.

Route tables (1/8) Info

Filter route tables

Name	Route table ID	Explicit subnet associat...	Edge associations	Main	VPC	Owner ID
-	rtb-027b77903074cdf80	-	-	Yes	vpc-0dbb652d74885e734	827738972389
Public Route Table 2	rtb-03db475a2412da960	subnet-0044316169d02...	-	No	vpc-0eed0b872611e50ee   La...	827738972389
Public Route Table 1	rtb-085909aa4ca781d91	subnet-0fee4c8ae9dd5...	-	No	vpc-0eed0b872611e50ee   La...	827738972389
Private Route Table 3	rtb-0a6863c70ceea6674	subnet-0783cd8e00df7...	-	No	vpc-0eed0b872611e50ee   La...	827738972389
Private Route Table 2	rtb-0cf83208a264ae39f	subnet-077e61fafc2b06...	-	No	vpc-0eed0b872611e50ee   La...	827738972389

#### rtb-0cf83208a264ae39f / Private Route Table 2

Details Routes Subnet associations Edge associations Route propagation Tags

#### Routes (2)

Filter routes Both

Destination	Target	Status
0.0.0.0/0	nat-0c9fe5355805f48df	Active
10.0.0.0/16	local	Active

This route table routes to the created **NAT gateway**, and associates with **Private subnet 2**. Now, private subnet 2 can access the Internet.

Route tables (1/8) Info

Filter route tables

Name	Route table ID	Explicit subnet associat...	Edge associations	Main	VPC	Owner ID
-	rtb-027b77903074cdf80	-	-	Yes	vpc-0dbb652d74885e734	827738972389
Public Route Table 2	rtb-03db475a2412da960	subnet-0044316169d02...	-	No	vpc-0eed0b872611e50ee   La...	827738972389
Public Route Table 1	rtb-085909aa4ca781d91	subnet-0fee4c8ae9dd5...	-	No	vpc-0eed0b872611e50ee   La...	827738972389
Private Route Table 3	rtb-0a6863c70ceea6674	subnet-0783cd8e00df7...	-	No	vpc-0eed0b872611e50ee   La...	827738972389
Private Route Table 2	rtb-0cf83208a264ae39f	subnet-077e61fafc2b06...	-	No	vpc-0eed0b872611e50ee   La...	827738972389
-	rtb-00cc9136b266d45d8	-	-	Yes	vpc-0eed0b872611e50ee   La...	827738972389

#### rtb-0cf83208a264ae39f / Private Route Table 2

Details Routes Subnet associations Edge associations Route propagation Tags

#### Explicit subnet associations (1)

Find subnet association

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
Private Subnet 2	subnet-077e61fafc2b068ea	10.0.3.0/24	-

## Task 3: Creating a bastion host instance in a public subnet

In this lab, we need an EC2 instance called **Bastion Host** to ssh into the scaling instances and increase its CPU load. This will help to test if Auto Scaling can automatically add more instances to handle the high CPU load. The instance configurations are basic, we should only pay attention to its **Security group**. The inbound rule of its security group allows **SSH**, which is necessary to ssh into the instance by PuTTY step 8.

Services Search [Alt+S] N. Virginia voclabs/user2362572-105488557@

Name: Bastion Host Add additional tags

Application and OS Images (Amazon Machine Image) Info

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

Search your full catalog including 1000s of application and OS images

Recents My AMIs Quick Start

Amazon Linux macOS Ubuntu Windows Red Hat

Amazon Machine Image (AMI)

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type

ami-04581fb744a7d11f (64-bit (x86)) / ami-0ef1cf6b946f12499 (64-bit (arm))

Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible

Summary

Number of instances: 1

Software Image (AMI)

Amazon Linux 2023 AMI 2023.0.2...read more

ami-00c39f71452c08778

Virtual server type (instance type)

t2.micro

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

Cancel Launch instance

VPC - required [Info](#)

vpc-0eed0b872611e50ee (Lab VPC)

Subnet [Info](#)

subnet-0fee4c8ae9dd5817 Public Subnet 1

Auto-assign public IP [Info](#)

Enable

Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☒ Create security group ☐ Select existing security group

Security group name - required

BastionSG

Description - required [Info](#)

Bastion Host security group

Inbound security groups rules

Security group rule 1 (TCP, 22, 1.53.126.195/32)

Type [Info](#) Protocol [Info](#) Port range [Info](#)

ssh TCP 22

Source type [Info](#) Name [Info](#) Description - optional [Info](#)

My IP Add CIDR, prefix list or security e.g. SSH for admin desktop

Summary

Number of instances [Info](#)

1

Software Image (AMI)

Amazon Linux 2 Kernel 5.10 AMI...read more

Virtual server type (instance type)

t2.micro

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

Cancel Launch instance

Create a new security group that allows SSH traffic

## Task 4: Creating a launch template

In this lab, I created a **Launch template** to prepare for the Auto Scaling Group. It was created from the AMI of the **CafeWebAppServer** instance, and it works quite similarly to the **Launch Configurations** I did in the previous labs. Both will determine how the additional instances generated by the Auto Scaling Group work.

EC2 > Launch templates > Create launch template

Create launch template

Creating a launch template allows you to create a saved instance configuration that can be reused, shared and launched at a later time. Templates can have multiple versions.

Launch template name and description

Launch template name - required

CafeWebserverTemplate

Template version description

a template

Summary

Software Image (AMI)

Cafe WebServer Image

Virtual server type (instance type)

t2.micro

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

Cancel Create launch template

Services Search [Alt+S]

Recents My AMIs Quick Start

☐ Don't include in launch template ☒ Owned by me ☐ Shared with me

Amazon Machine Image (AMI)

Cafe WebServer Image

ami-0fa0568da9849a30c

Summary

Software Image (AMI)

Cafe WebServer Image

Virtual server type (instance type)

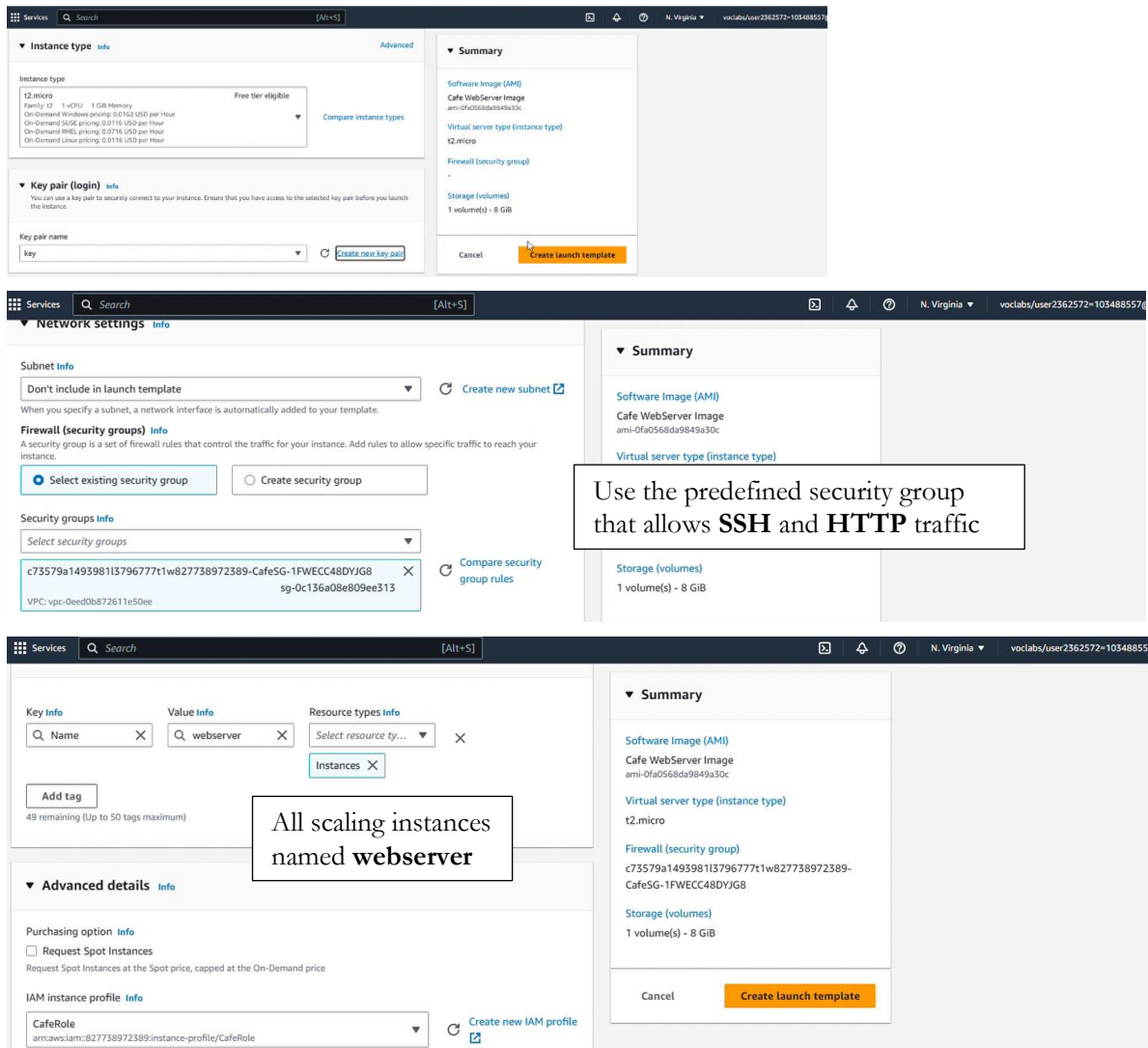
t2.micro

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB



## Task 5: Creating an Auto Scaling group

Now, everything is ready for the Auto Scaling Group. It will hold the responsibility to automate the process of checking the current state of the system and add more instances when needed.

### Important points:

- The scaling instances will be launched in 2 Private subnets
- The group size is:
  - o Desired capacity: 2
  - o Minimum capacity: 2
  - o Maximum capacity: 6
- A target tracking policy is included

[Alt+S] N. Virginia voclabs/user2362572-103488557

uto 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.

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.

[Create a launch template](#)

Select the **Launch template** created in task 4

[Alt+S] N. Virginia voclabs/user2362572-103488557

ito Scaling group

### Choose instance launch options [Info](#)

Choose the VPC network environment that your instances are launched into, and customize the instance types and purchase options.

**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.

[Create a VPC](#)

Availability Zones and subnets

Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.

Select Availability Zones and subnets

us-east-1a | subnet-0c0a03e8984a7b406 (Private Subnet 1) 10.0.2.0/24

us-east-1b | subnet-077e61fafc2b068ea (Private Subnet 2) 10.0.3.0/24

The scaling instances based on this template will be created in **Private subnets**

[Alt+S] N. Virginia voclabs/user2362572-103488557

### Configure group size and scaling policies - *optional* [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

Minimum capacity

Maximum capacity

Define the **group size** (the scaling range)

Scaling policies - optional

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  
25

Instances need  
60 seconds warm up before including in metric

Scaling policy is the metric for the CloudWatch alarms that invoke the scaling process.

**Result:** 2 new instances were added.

[Alt+S]

N. Virginia

voclabs/user2362572-103488557@

Instances (4) Info

Find instance by attribute or tag (case-sensitive)

Connect

Instance state

Actions

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
<input type="checkbox"/>	CafeWebAppServer	i-073f0d3bf5160e0f2	Running	t2.micro	2/2 checks passed	User: arn:aws:us-east-1a	us-east-1a	ec2-54-235-45-1
<input type="checkbox"/>	Bastion Host	i-0f83d9a00441199a2	Running	t2.micro	2/2 checks passed	User: arn:aws:us-east-1a	us-east-1a	ec2-3-81-147-14
<input type="checkbox"/>	webserver	i-054411e12fd06e158	Running	t2.micro	Initializing	User: arn:aws:us-east-1a	us-east-1a	-
<input type="checkbox"/>	webserver	i-088404e702914cf10	Running	t2.micro	Initializing	User: arn:aws:us-east-1b	us-east-1b	-

## Task 6: Creating a load balancer

Before creating a Load Balancer, I need to create a **Security Group** and a **Target Group** to use on the Load Balancer configuration page.

Security group: The Load Balancer security group is named **ELBSG**, and it allows **HTTP** access from anywhere.

EC2 > Security Groups > Create security group

### Create security group [Info](#)

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 [Info](#)  
CafeELBSG  
Name cannot be edited after creation.

Description [Info](#)  
Enable web access to load balancer

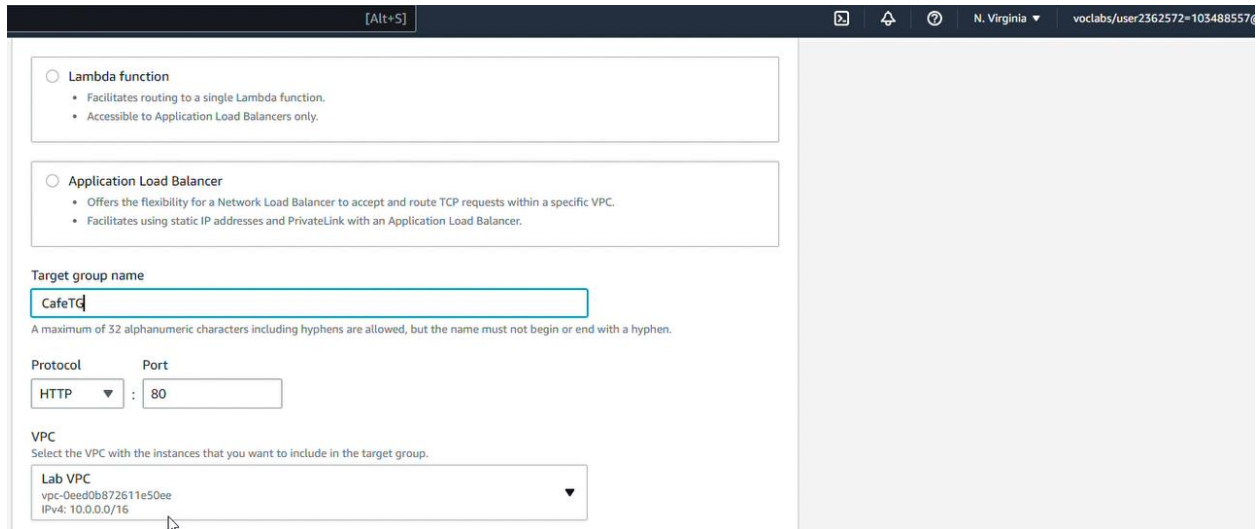
VPC [Info](#)  
vpc-0eed0b872611e50ee

**Inbound rules [Info](#)**

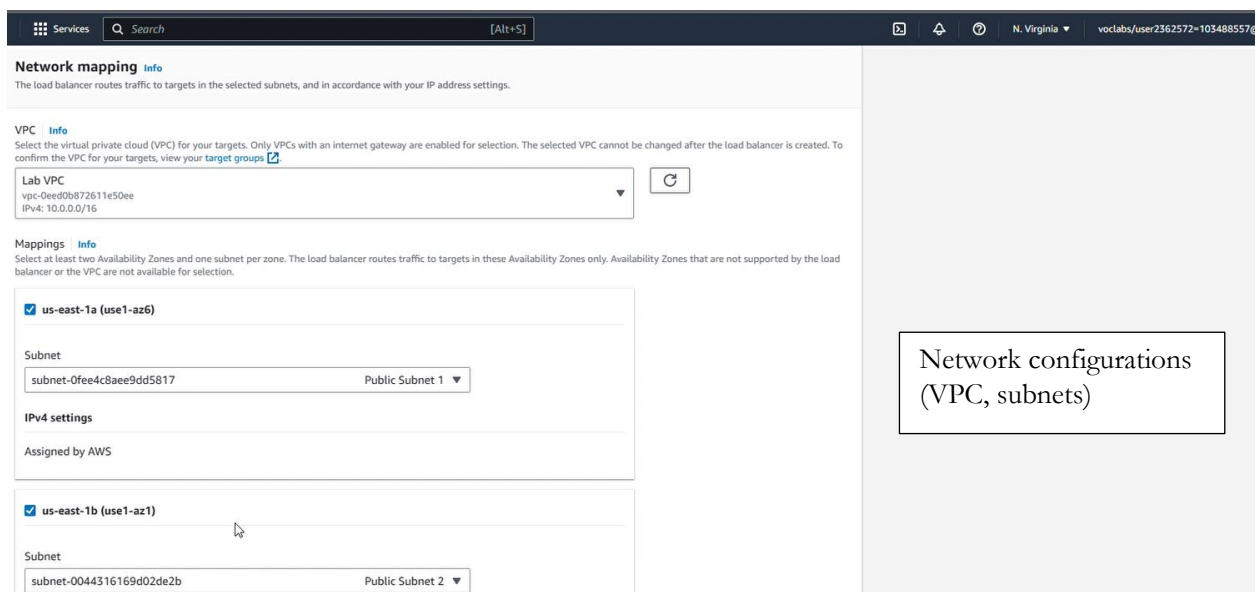
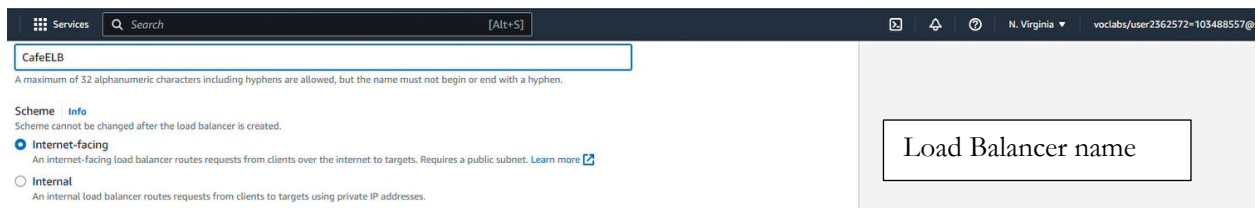
Type <a href="#">Info</a>	Protocol <a href="#">Info</a>	Port range <a href="#">Info</a>	Source <a href="#">Info</a>	Description - optional <a href="#">Info</a>
HTTP	TCP	80	Anywhere... 0.0.0.0	



Target Group: I only created a Target group with a name and attached it to my VPC. I did not register targets for it yet because it would have the targets when it had been attached to the Load Balancer and the Auto Scaling Group.



Load Balancer: used 2 public subnets and used the **ELBSG** Security Group as well as the **CafeTG** Target Group specified above. This balancer was also attached to my VPC.



Services Search [Alt+S] N. Virginia voclabs/user2362572=103488557@

Security groups

Select up to 5 security groups

Create new security group

CafeELBSG sg-0bccfe66a474127d5 VPC: vpc-0eed0b872611e50ee

**Listeners and routing** Info

A listener is a process that checks for connection requests using the port and protocol you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets.

▼ Listener HTTP:80 Remove

Protocol HTTP Port 80 1-65535

Default action Info

Forward to CafeTG HTTP

Target type: Instance, IPv4

Create target group

Security group and Target group

## The configurations summary

Services Search [Alt+S] N. Virginia voclabs/user2362572=103488557@

applications. Additional charges apply

► **Tags - optional**

Consider adding tags to your load balancer. Tags enable you to categorize your AWS resources so you can more easily manage them. The 'Key' is required, but 'Value' is optional. For example, you can have Key = production-webserver, or Key = webserver, and Value = production.

**Summary**

Review and confirm your configurations. Estimate cost

<p><b>Basic configuration</b> Edit</p> <p>CafeELB</p> <ul style="list-style-type: none"> <li>Internet-facing</li> <li>IPv4</li> </ul>	<p><b>Security groups</b> Edit</p> <ul style="list-style-type: none"> <li>CafeELBSG sg-0bccfe66a474127d5</li> </ul>	<p><b>Network mapping</b> Edit</p> <p>VPC vpc-0eed0b872611e50ee</p> <p>Lab VPC</p> <ul style="list-style-type: none"> <li>us-east-1a subnet-0fee4c8aee9dd5817 Public Subnet 1</li> <li>us-east-1b subnet-0044316169d02de2b Public Subnet 2</li> </ul>	<p><b>Listeners and routing</b> Edit</p> <ul style="list-style-type: none"> <li>HTTP:80 defaults to CafeTG</li> </ul>
---	---	---	---

I waited for a few minutes until the Load Balancer status changed to **‘Active’**. That means it was ready to be attached to an Auto Scaling Group. I returned to the **Auto Scaling Group** → Select the group → **Edit** → Select **CafeELB** as the load balancer.

[Alt+S] N. Virginia voclabs/user2362572=103488557@

**Load balancing - optional**

**Load balancers**

☒ Application, Network or Gateway Load Balancer target groups

Only instance target groups that belong to the same VPC as your Auto Scaling group are available for selection.

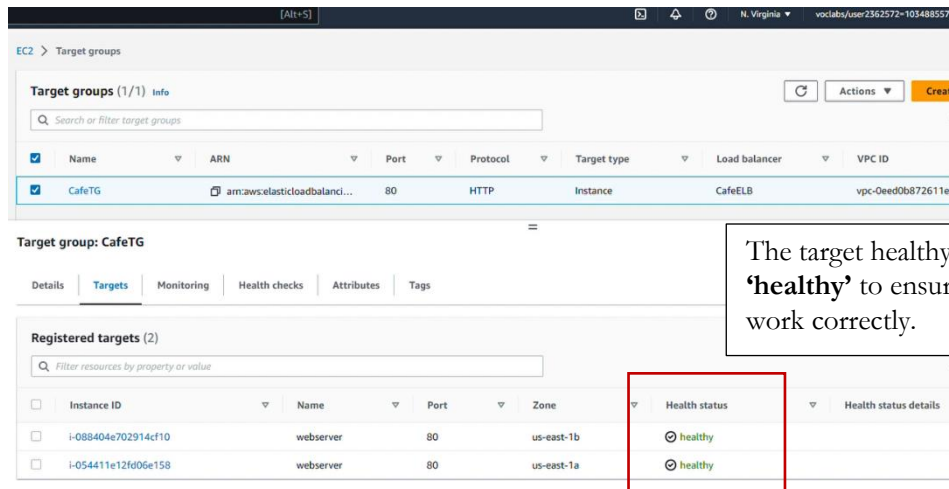
Select target groups

CafeTG | HTTP Application Load Balancer: CafeELB



## Task 7: Testing the web application

I tested all my configurations by accessing the web application through the **Load Balancer DNS Name** instead of using the EC2 Public IP address. All targets in the Target Group must be **'healthy'** before accessing the web.

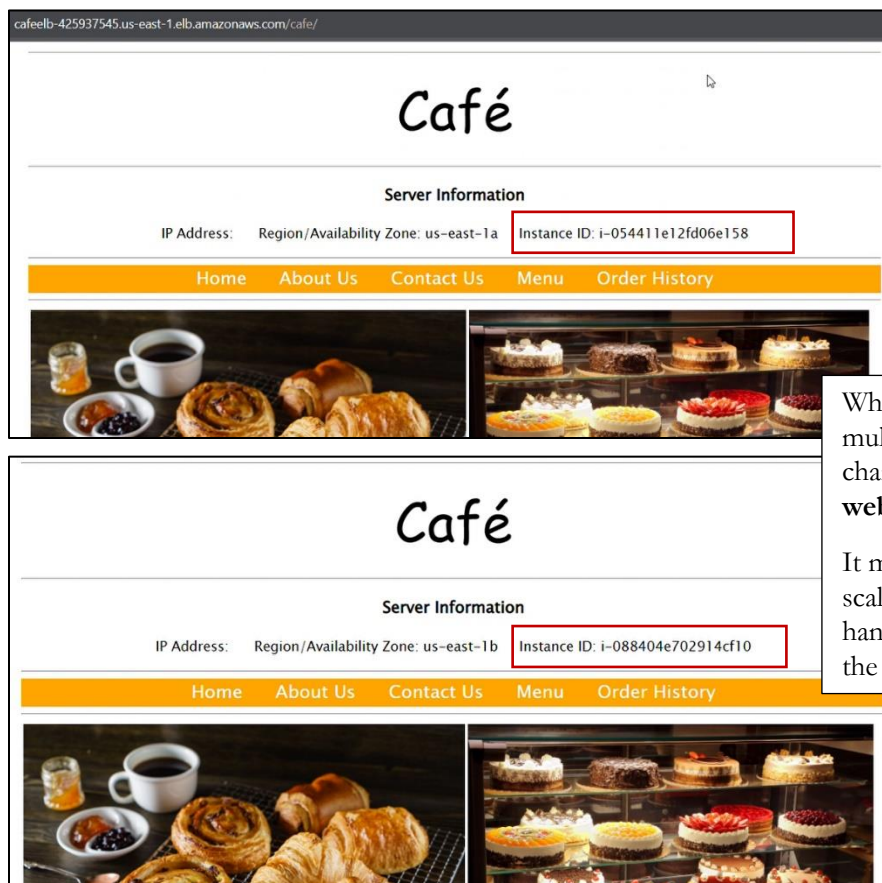


The screenshot shows the AWS Management Console for Target Groups. The 'CafeTG' target group is selected, and its 'Targets' tab is active. Two targets are listed, both with a 'healthy' status, indicated by a green checkmark icon. A red box highlights the 'Health status' column.

Instance ID	Name	Port	Zone	Health status	Health status details
i-088404e702914cf10	webserver	80	us-east-1b	healthy	
i-054411e12fd06e158	webserver	80	us-east-1a	healthy	

The target healthy status must be **'healthy'** to ensure that it can work correctly.

## Testing result



The first screenshot shows the Café website with the Instance ID 'i-054411e12fd06e158' highlighted in the Server Information section. The second screenshot shows the same website after a refresh, with the Instance ID 'i-088404e702914cf10' highlighted. This demonstrates that the website is using different webserver instances to handle access.

When I refreshed the website multiple times, the Instance ID changed between two **webserver** instances.

It means the **webserver** scaling instances alternately handle the access to increase the web **availability**.

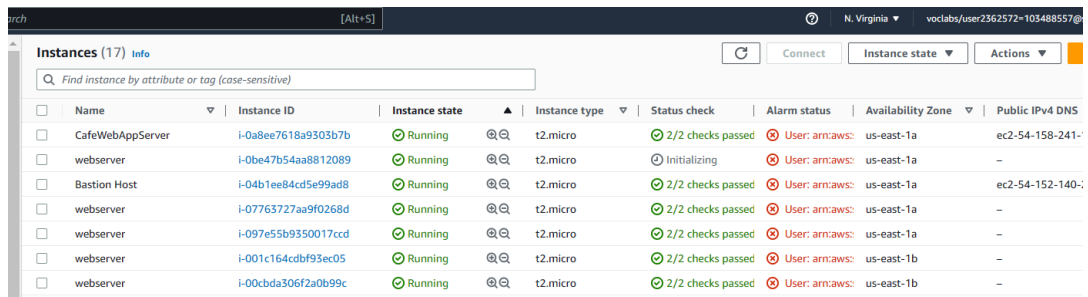
## Task 8: Testing automatic scaling under load

In this task, I increased the load on the web server to see whether the web app can scale out automatically.

I ssh into the Bastion Host, then I ssh into a random web server. I need to use the Bastion Host because all scaling instances are stored in the private subnets. After accessing the web server instance, I run the provided commands on Canvas to increase the CPU load.

### Result

The app scaled out by adding many instances:



<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
<input type="checkbox"/>	CafeWebAppServer	i-0a8ee7618a9303b7b	Running	t2.micro	2/2 checks passed	User: arm:aws: us-east-1a	us-east-1a	ec2-54-158-241-
<input type="checkbox"/>	webserver	i-0be47b54aa8812089	Running	t2.micro	Initializing	User: arm:aws: us-east-1a	us-east-1a	-
<input type="checkbox"/>	Bastion Host	i-04b1ee84cd5e99ad8	Running	t2.micro	2/2 checks passed	User: arm:aws: us-east-1a	us-east-1a	ec2-54-152-140-
<input type="checkbox"/>	webserver	i-07763727aa9f0268d	Running	t2.micro	2/2 checks passed	User: arm:aws: us-east-1a	us-east-1a	-
<input type="checkbox"/>	webserver	i-097e55b9350017ccd	Running	t2.micro	2/2 checks passed	User: arm:aws: us-east-1a	us-east-1a	-
<input type="checkbox"/>	webserver	i-001c164cbbf93ec05	Running	t2.micro	2/2 checks passed	User: arm:aws: us-east-1b	us-east-1b	-
<input type="checkbox"/>	webserver	i-00cbda306f2a0b99c	Running	t2.micro	2/2 checks passed	User: arm:aws: us-east-1b	us-east-1b	-