

Assignment 4 – ALB and ASG

ACCEPTANCE CRITERIA – Include the followings in the PDF:

- 2 links for ALB and NLB.
- Screenshot of healthy instances in both TGs of ALB and NLB.
- Screenshot of the Activity tab in the ASG.

Tasks:

Task 1. Run 2 web servers behind ALB

- a. Create an SG for the ALB that allows access from the internet on port 80 (HTTP). Give a meaningful name like “alb-sg”. The meaningful name will help when whitelisting this SG in the web servers’ SG.
- b. Create an SG for an EC2 instance (web servers). Open up port 80 from the ALB SG. That means the web servers only allow access from the load balancer.
- c. Create 2 web servers in us-east-1a and us-east-1b AZs with different HTML content. To do that, hit “Edit” in the “Network Settings” and select subnets with “**us-east-1a**” for the first instance and “**us-east-1b**” for the second instance. It is important because these AZs are where your load balancer nodes will be created. Lets say you created 2 instances in AZ 1a and 1b. But your load balancer nodes are created in AZ 1d and 1f, the load balancer cannot route the requests to the servers and you will see “**unused**” state in the target group.
- d. Select the SG for the webserver you created in the previous step.
- e. Put the following script in “User Data”. So, your web server starts automatically when the server starts.

```
#!/bin/bash
yum install -y httpd
systemctl start httpd
systemctl enable httpd
echo "<h1>Hello YourName from $(hostname -f)</h1>" > /var/www/html/index.html
```

#!/bin/bash – is equivalent to “sudo -s” in bash.

- f. Create ALB. Select **us-east-1a** and **us-east-1b** AZs for HA (High Availability). Create the TG and register the servers. And select the TG you created.

Task 2. Run web servers behind NLB

Similar to launching an ALB. The only difference is, to change TG protocol to **TCP** (Layer 4), not HTTP (Layer 7).

Task 3. Run the web server in ASG

- a. Deregister instances behind the ALB. We will register them through ASG. So they can scale automatically.

- b. Create a launch template. Not launch configuration because the launch template is recommended. Launch template allows you to select AMI like EC2 where launch configuration requires you to enter AMI ID.
 - i. Give it a name
 - ii. Select the Amazon Linux AMI.
 - iii. Select instance type, t2.micro.
 - iv. Expand advanced. Select the IAM profile. Just in case you want to debug your web app, for example, to see if the web server is up with a custom HTML. But the web app is already configured automatically with user data.
 - v. Enter the previous User Data above.
 - vi. Select the web server's SG. Created in task 1.
 - vii. Select any key pair. It doesn't matter. Because we use Session Manager to SSH into the instance if needed.
- c. Create the Auto Scaling Group.
 - i. Select launch template/configuration.
 - ii. Select AZs (Subnets). That is where your instances launched.
 - iii. Click on attach to an existing load balancer and select the default TG of the ALB.
 - iv. Select ELB in the health checks panel.
 - v. Set desired, min, and max capacity. Set a target tracking scale policy.
- d. Mimic the high CPU utilization with the "stress" library to test scaling out behavior. See the last page for further reference.

Task 4. Clean up ALB, NLB, EC2 instances. They cost huge amounts.

Step by step

Task 1 - Run 2 web servers behind ALB

Create Security Groups for ALB

- Create an SG for the ALB which is open to the world.
- Create an SG for web servers that allows ALB's SG

Create Application Load Balancer Security Group (Outbound Rule is Default - All Traffic)

Security group name: my-lab-alb-sg
Security group ID: sg-03e5e025e377518eb
Description: Lab Application Load Balancer Security Group
VPC ID: vpc-0b978358e22761686
Owner: 409673912482
Inbound rules count: 1 Permission entry
Outbound rules count: 1 Permission entry

Inbound rules (1/1)

Type	Protocol	Port range	Source	Description
HTTP	TCP	80	0.0.0.0/0	-

Create EC2 Web Server Security Group (Outbound Rule is Default - All Traffic)

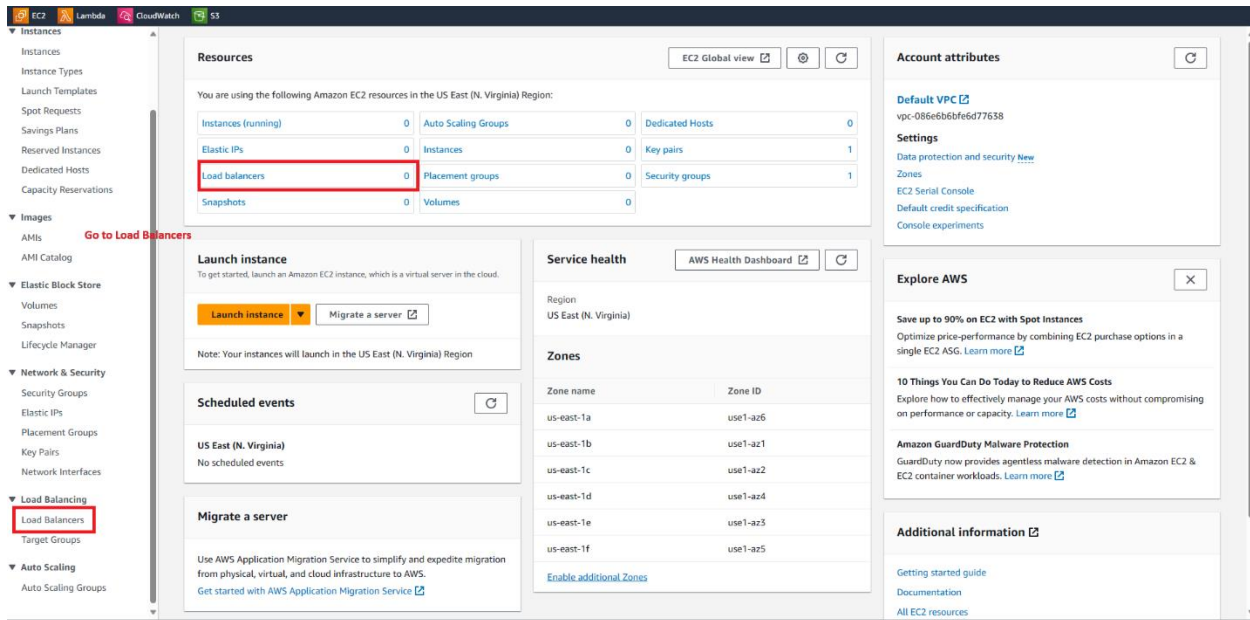
Security group name: my-lab-EC2-Server-sg
Security group ID: sg-0a370c15c5b405b61
Description: Web Server Security Group
VPC ID: vpc-0b978358e22761686
Owner: 409673912482
Inbound rules count: 1 Permission entry
Outbound rules count: 1 Permission entry

Inbound rules (1/1)

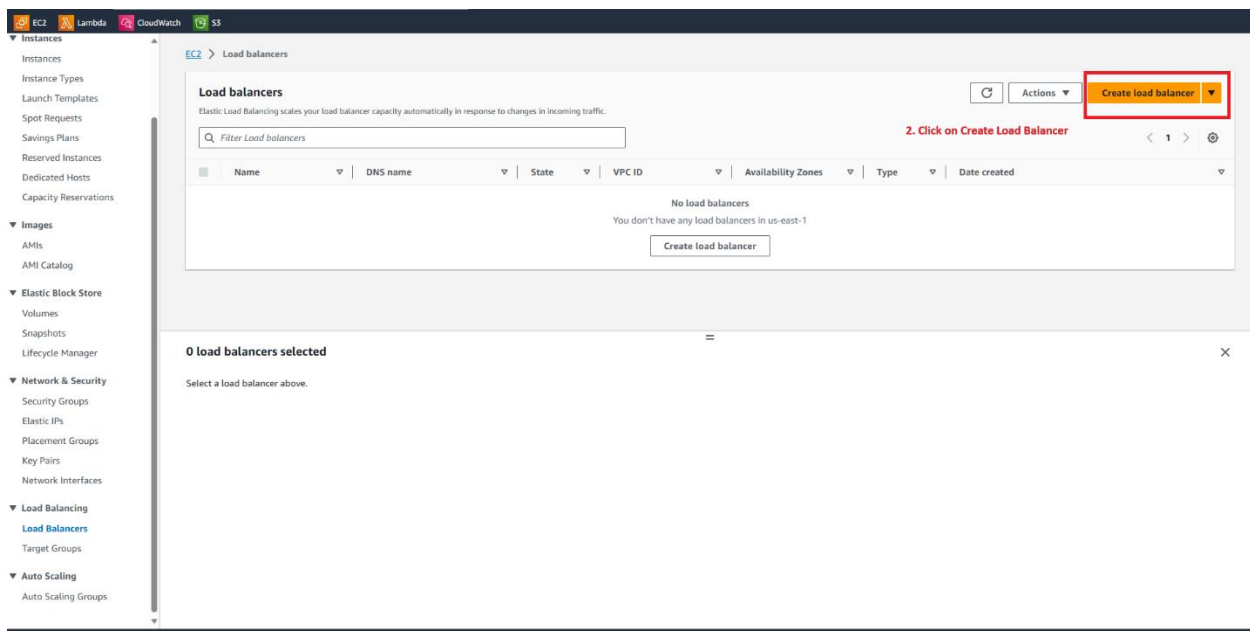
Type	Protocol	Port range	Source	Description
HTTP	TCP	80	sg-03e5e025e377518eb	-

Create an ALB

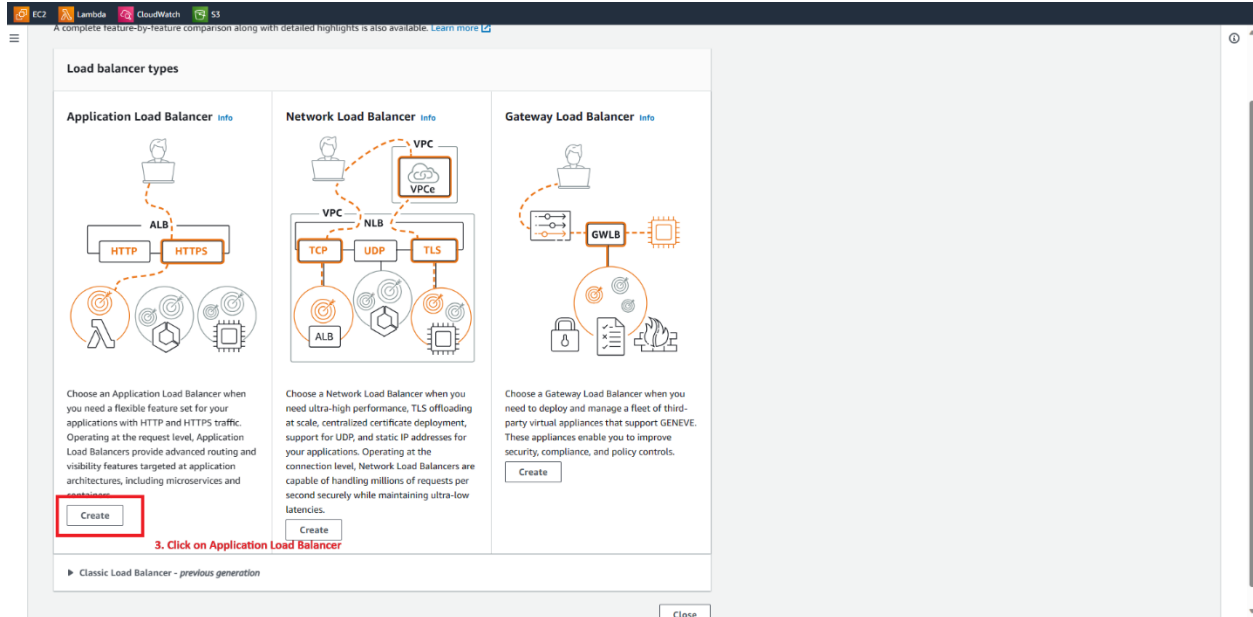
1: Go to Load Balancers Display in EC2 Dashboard.



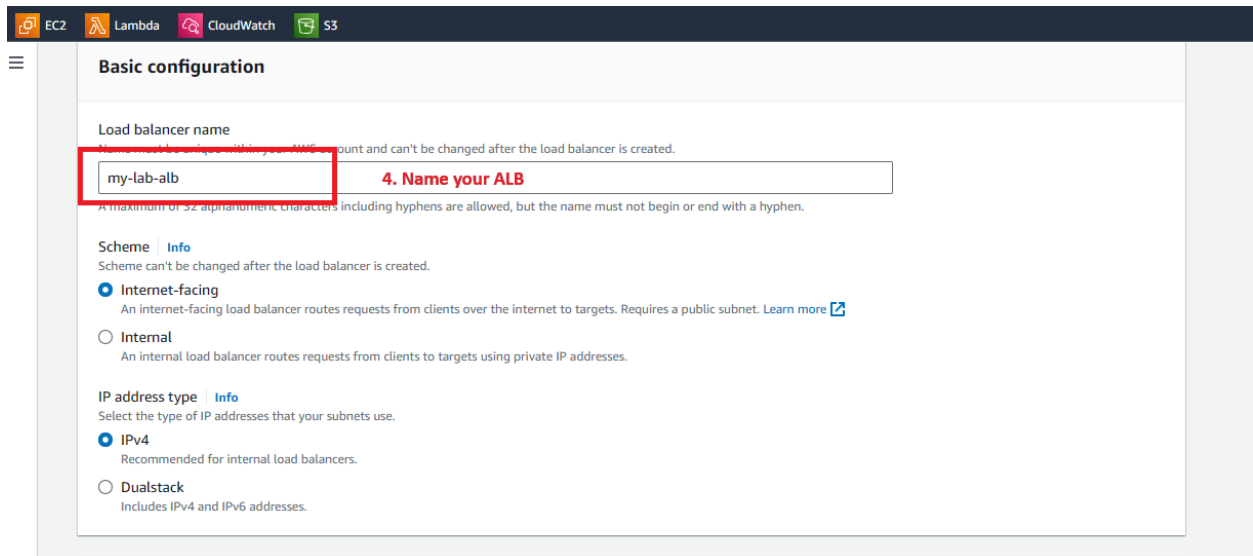
2: Once in Load Balancers Display, click on Create Load Balancer.



3: Click on Create Application Load Balancer.



4: Name your ALB.



5: Select VPC

6: Select at least 2 AZ zones/subnets

The screenshot shows the AWS Network mapping console. At the top, there are service icons for EC2, Lambda, CloudWatch, and S3. The main heading is "Network mapping" with an "Info" link. Below the heading, a description states: "The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings."

The "VPC" section is highlighted with a red box. It includes an "Info" link and a description: "Select the virtual private cloud (VPC) for your targets or you can [create a new VPC](#). The selected VPC can't be changed after the load balancer is created. To confirm the VPC for your targets, view your [target groups](#)." Below this is a dropdown menu showing "vpc-086e6b6bfe6d77638" with an IPv4 address of "172.31.0.0/16". A refresh button is to the right.

The "Mappings" section is also highlighted with a red box. It includes an "Info" link and a description: "Select at least two Availability Zones and one subnet per zone. The load balancer routes traffic to targets in these Availability Zones only. Availability Zones that are not supported by the load balancer or the VPC are not available for selection."

Two mappings are listed, each with a red box around the "us-east-1a (use1-az6)" and "us-east-1b (use1-az1)" labels. The first mapping shows a subnet dropdown with "subnet-0b348d5147a01b3bb" and a warning message: "The selected subnet is not a private subnet. This means that your internal load balancer can receive internet traffic. You can proceed with this selection; however, to prevent internet traffic from reaching your load balancer, you must choose a private subnet or update this subnet's route table in the [VPC console](#)." The IPv4 address is "Assigned from CIDR 172.31.32.0/20".

The second mapping shows a subnet dropdown with "subnet-09d9225a6915e52e8" and the same warning message. To the right of this mapping, the text "5. Choose VPC and 2 AZ" is visible.

7: Select ALB SG you created

The screenshot shows the AWS Security groups console. The heading is "Security groups" with an "Info" link. Below the heading, a description states: "A security group is a set of firewall rules that control the traffic to your load balancer. Select an existing security group, or you can [create a new security group](#)."

The "Security groups" section is highlighted with a red box. It includes a dropdown menu with the text "Select up to 5 security groups". Below the dropdown is a list of security groups, with "my-lab-alb-sg" selected. The list also shows the ID "sg-0c7f375ccf041fbc" and the VPC "vpc-086e6b6bfe6d77638". A close button (X) is to the right of the selected group. A refresh button is to the right of the dropdown.

8: Select TG you created

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

Port

HTTP ▼

:

80

1-65535

Default action

Info

Forward to

TG1

▼

Target type: Lambda, IPv4

Create target group [↗](#)

↻

Listener tags - optional

Consider adding tags to your listener. Tags enable you to categorize your AWS resources so you can more easily manage them.

Add listener tag

You can add up to 50 more tags.

Add listener

9: Create your load balancer.

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

Port

HTTP ▼

:

80

1-65535

Default action

Info

Forward to

TG1

▼

Target type: Lambda, IPv4

Create target group [↗](#)

↻

Listener tags - optional

Consider adding tags to your listener. Tags enable you to categorize your AWS resources so you can more easily manage them.

Add listener tag

You can add up to 50 more tags.

Add listener

Task 2 – Create an NLB

1: Go to Load Balancers from EC2 dashboard and hit create Load Balancer.

The image shows two screenshots of the AWS Management Console. The top screenshot displays the EC2 dashboard with the 'Load balancers' link highlighted in the left-hand navigation menu. The right-hand pane shows account attributes and settings. The bottom screenshot shows the 'Load balancers' page, where the 'Create load balancer' button is highlighted in the top right corner. Below this, a message states 'No load balancers' and 'You don't have any load balancers in us-east-1'. A modal at the bottom indicates '0 load balancers selected'.

Resources

You are using the following Amazon EC2 resources in the US East (N. Virginia) Region:

Instances (running)	Auto Scaling Groups	Dedicated Hosts
0	0	0
Elastic IPs	Instances	Key pairs
0	0	1
Load balancers	Placement groups	Security groups
0	0	1
Snapshots	Volumes	
0	0	

Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

[Launch instance](#) [Migrate a server](#)

Note: Your instances will launch in the US East (N. Virginia) Region

Scheduled events

US East (N. Virginia)
No scheduled events

Migrate a server

Use AWS Application Migration Service to simplify and expedite migration from physical, virtual, and cloud infrastructure to AWS.
[Get started with AWS Application Migration Service](#)

Service health

[AWS Health Dashboard](#)

Region: US East (N. Virginia)

Zones

Zone name	Zone ID
us-east-1a	use1-az6
us-east-1b	use1-az1
us-east-1c	use1-az2
us-east-1d	use1-az4
us-east-1e	use1-az3
us-east-1f	use1-az5

[Enable additional Zones](#)

Account attributes

Default VPC
vpc-086e6bb6d77638

Settings
[Data protection and security](#)
[Zones](#)
[EC2 Serial Console](#)
[Default credit specification](#)
[Console experiments](#)

Explore AWS

Save up to 90% on EC2 with Spot instances
Optimize price-performance by combining EC2 purchase options in a single EC2 ASG. [Learn more](#)

10 Things You Can Do Today to Reduce AWS Costs
Explore how to effectively manage your AWS costs without compromising on performance or capacity. [Learn more](#)

Amazon GuardDuty Malware Protection
GuardDuty now provides agentless malware detection in Amazon EC2 & EC2 container workloads. [Learn more](#)

Additional information
[Getting started guide](#)
[Documentation](#)
[All EC2 resources](#)

Load balancers

Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

[Filter Load balancers](#)

2. Click on Create Load Balancer

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

[Create load balancer](#)

0 load balancers selected

Select a load balancer above.

2: Select Network Load Balancer and hit create.

The screenshot shows the AWS Management Console's 'Load balancer types' page. At the top, there's a navigation bar with icons for EC2, Lambda, CloudWatch, S3, IAM, and RDS. Below the navigation bar, the page title is 'Load balancer types'. There are three main sections, each with a diagram and a description:

- Application Load Balancer**: Diagram shows a client connecting to an ALB, which routes traffic to targets. Description: 'Choose an Application Load Balancer when you need a flexible feature set for your applications with HTTP and HTTPS traffic. Operating at the request level, Application Load Balancers provide advanced routing and visibility features targeted at application architectures, including microservices and containers.' A 'Create' button is at the bottom.
- Network Load Balancer**: Diagram shows a client connecting to an NLB, which routes traffic to targets. Description: 'Choose a Network Load Balancer when you need ultra-high performance, TLS offloading at scale, centralized certificate deployment, support for UDP, and static IP addresses for your applications. Operating at the connection level, Network Load Balancers are capable of handling millions of requests per second securely while maintaining ultra-low latency.' A 'Create' button is at the bottom, highlighted with a red box.
- Gateway Load Balancer**: Diagram shows a client connecting to a GWLB, which routes traffic to targets. Description: 'Choose a Gateway Load Balancer when you need to deploy and manage a fleet of third-party virtual appliances that support GENEVE. These appliances enable you to improve security, compliance, and policy controls.' A 'Create' button is at the bottom.

At the bottom left, there's a link for 'Classic Load Balancer - previous generation'. At the bottom right, there's a 'Close' button.

3: Name your NLB

The screenshot shows the AWS Management Console's 'Create Network Load Balancer' page. At the top, there's a navigation bar with icons for EC2, Lambda, CloudWatch, S3, IAM, and RDS. Below the navigation bar, the page title is 'Create Network Load Balancer'. There's a brief description of the Network Load Balancer. Below that, there's a section titled 'How Elastic Load Balancing works'. The main section is 'Basic configuration', which includes:

- Load balancer name**: A text input field containing 'my-lab-nlb', highlighted with a red box. Below the field, there's a note: 'A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.'
- Scheme**: A section with two radio buttons: 'Internet-facing' (selected) and 'Internal'. Below each radio button is a description of the scheme.
- IP address type**: A section with two radio buttons: 'IPv4' (selected) and 'Dualstack'. Below each radio button is a description of the IP address type.

4: Select VPC and 2 AZs

Network mapping [Info](#)

The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

VPC [Info](#)

Select the virtual private cloud (VPC) for your targets or you can [create a new VPC](#). The selected VPC can't be changed after the load balancer is created. To confirm the VPC for your targets, view your [target groups](#).

-
vpc-086e6b6bfe6d77638
IPv4: 172.31.0.0/16

Mappings [Info](#)

Select at least two Availability Zones and one subnet per zone. The load balancer routes traffic to targets in these Availability Zones only. Availability Zones that are not supported by the load balancer or the VPC are not available for selection.

☒ **us-east-1a (use1-az6)**

Subnet
subnet-0b348d5147a01b3bb

⚠ The selected subnet is not a private subnet. This means that your internal load balancer can receive internet traffic.
You can proceed with this selection; however, to prevent internet traffic from reaching your load balancer, you must choose a private subnet or update this subnet's route table in the [VPC console](#).

IPv4 address
Assigned from CIDR 172.31.32.0/20

☒ **us-east-1b (use1-az1)**

5. Choose VPC and 2 AZ

Subnet
subnet-09d9225a6915e52e8

⚠ The selected subnet is not a private subnet. This means that your internal load balancer can receive internet traffic.
You can proceed with this selection; however, to prevent internet traffic from reaching your load balancer, you must choose a private subnet or update this subnet's route table in the [VPC console](#).

5: Select SG

Security groups [Info](#)

A security group is a set of firewall rules that control the traffic to your load balancer. Select an existing security group, or you can [create a new security group](#).

Security groups

Select up to 5 security groups

my-lab-alb-sg
sg-0c7f375ccf041fbc8 VPC: vpc-086e6b6bfe6d77638

6: Select TG (remember, the protocol for NLB is TCP)

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 TCP:80 [Remove](#)

Protocol **Port** **Default action** [Info](#)

TCP 80 Forward to TG1 Target type: Instance, IPv4 TCP [Refresh](#)

[Create target group](#)

Listener tags - optional

Consider adding tags to your listener. Tags enable you to categorize your AWS resources so you can more easily manage them.

[Add listener tag](#)

You can add up to 50 more tags.

[Add listener](#)

Add-on services - optional

Additional AWS services can be integrated with this load balancer at launch. You can also add these and other services after your load balancer is created by reviewing the "Integrated Services" tab for the selected load balancer.

AWS Global Accelerator [Info](#)

☐ Create an accelerator to get static IP addresses and improve the performance and availability of your applications. [Additional charges apply](#)

Load balancer 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

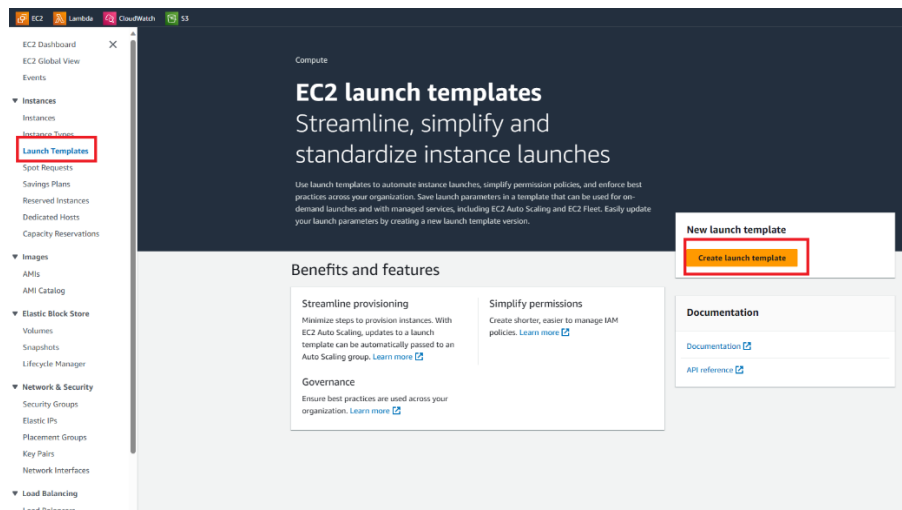
© 2023, Amazon Web Services

7: Create a Load Balancer.

8: Follow the steps in Part D of this assignment to create 2 links for app1 and app2. Using rules and condition

Task 3 – Run the Web Server behind the ALB in ASG

1: Go to Launch Template in EC2 dashboard and hit create a launch template



2: Provide a name and select guidance for a detailed assistance

EC2

Lambda

CloudWatch

S3

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

my-lab-server

Must be unique to this account. Max 128 chars. No spaces or special characters like '&', '\', '@'.

Template version description

A prod webserver for MyApp

Max 255 chars

Auto Scaling guidance [Info](#)

Select this if you intend to use this template with EC2 Auto Scaling

☒ Provide guidance to help me set up a template that I can use with EC2 Auto Scaling

▶ Template tags

▶ Source template

Launch template contents

Specify the details of your launch template below. Leaving a field blank will result in the field not being included in the launch template.

▼ Application and OS Images (Amazon Machine Image) - required [Info](#)

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

▼ Summary

Software Image (AMI)
-

Virtual server type (instance type)
-

Firewall (security group)
-

Storage (volumes)
-

Free tier: In your first year includes

750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

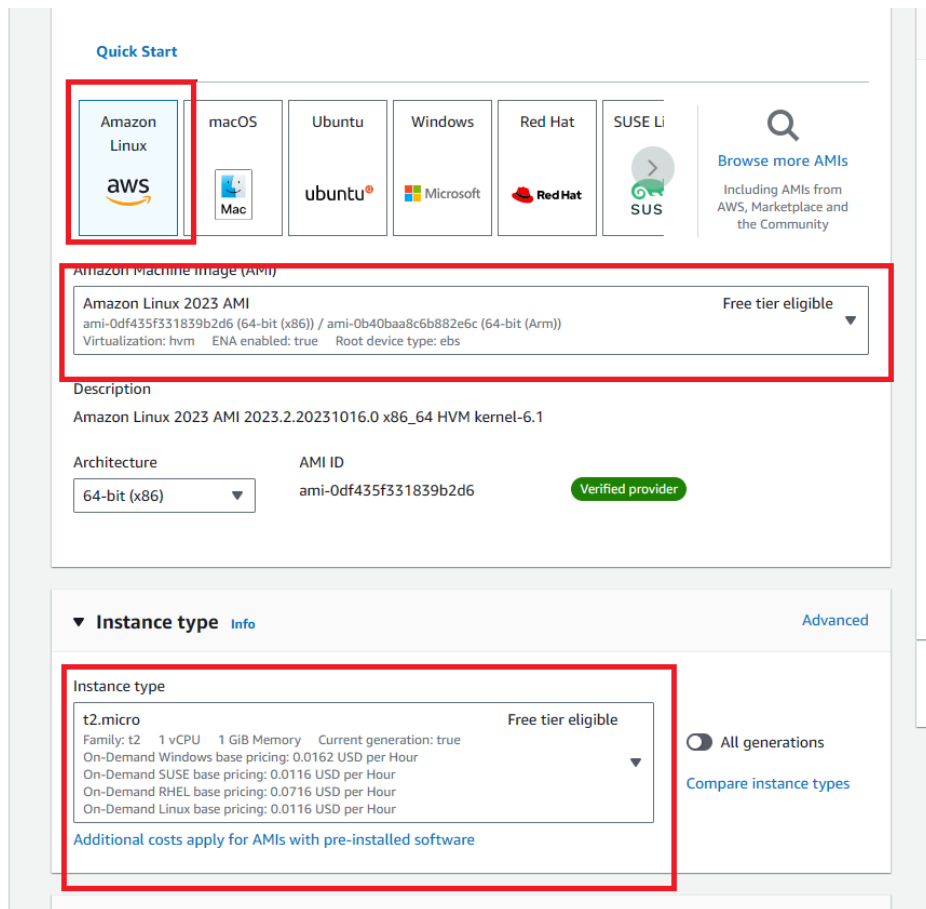
×

Cancel

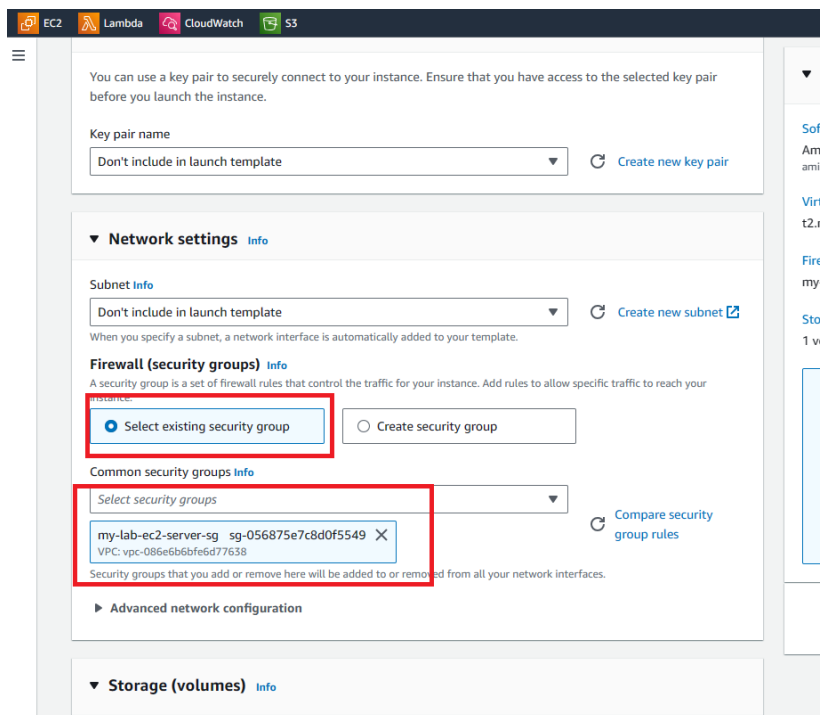
Create launch template

3: Select AMI

4: Select Instance Type



5: Select the security group you created. Then hit create.



Create an Auto-Scaling Group

1: From EC2 dashboard, go to Auto Scaling Groups and click on Create ASG

The screenshot displays the Amazon EC2 console interface. On the left, a navigation sidebar lists various services, with 'Auto Scaling' and 'Auto Scaling Groups' highlighted. The main content area features a large header for 'Amazon EC2 Auto Scaling' with a sub-header 'helps maintain the availability of your applications'. Below this, a 'Create Auto Scaling group' button is prominently displayed. A diagram titled 'How it works' illustrates the scaling process, showing a group of instances with labels for 'Minimum size', 'Desired capacity', 'Maximum size', and 'Scale out as needed'. To the right, sections for 'Pricing' and 'Getting started' are visible, providing additional information and links for users.

Amazon EC2 Auto Scaling
helps maintain the availability of your applications

Auto Scaling groups are collections of Amazon EC2 instances that enable automatic scaling and fleet management features. These features help you maintain the health and availability of your applications.

Create Auto Scaling group

Get started with EC2 Auto Scaling by creating an Auto Scaling group.

[Create Auto Scaling group](#)

How it works

An Auto Scaling group is a collection of Amazon EC2 instances that are treated as a logical unit. You configure settings for a group and its instances as well as define the group's minimum, maximum, and desired capacity. Setting different minimum and maximum capacity values forms the bounds of the group, which allows the group to scale as the load on your application spikes higher or lower, based on demand. To scale the Auto Scaling group, you can either make manual adjustments to the desired capacity or let Amazon EC2 Auto Scaling automatically add and remove

Pricing

Amazon EC2 Auto Scaling features have no additional fees beyond the service fees for Amazon EC2, CloudWatch (for scaling policies), and the other AWS resources that you use. Visit the pricing page of each service to learn more.

Getting started

[What is Amazon EC2 Auto Scaling?](#)

[Getting started with Amazon EC2 Auto Scaling](#)

[Set up a scaled and load-balanced application](#)

[FAQ](#)

2: Name your group and select your Launch template you previously created.

EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1
Choose launch template or configuration

Step 2
Configure settings

Step 3 (optional)
Configure advanced options

Step 4 (optional)
Configure group size and scaling policies

Step 5 (optional)
Add notifications

Step 6 (optional)
Add tags

Step 7
Review

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.

my-lab-as-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.

my-lab-server

[Create a launch template](#)

Version
Default (1)

[Create a launch template version](#)

Description	Launch template	Instance type
-	my-lab-server	t2.micro
AMI ID	lt-0ca69b78349a3fb83	
ami-0dc2d3e4c0f9ebd18	Security groups	Request Spot Instances
	-	No
Key pair name	Security group IDs	
-	sg-0a370c15c5b405b61	

Additional details

Storage (volumes)	Date created
-	Sun Jul 11 2021 11:24:24 GMT-0500 (Central Daylight Time)

Cancel

Next

2) Name Group

3) Select Your Launch Template

4) Click Next

Step 1
Choose launch template or configuration

Step 2
Configure settings

Step 3 (optional)
Configure advanced options

Step 4 (optional)
Configure group size and scaling policies

Step 5 (optional)
Add notifications

Step 6 (optional)
Add tags

Step 7
Review

Configure settings [Info](#)

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

Instance purchase options [Info](#)

Use the launch template to create a uniform configuration among all of the instances in the group. Or define options to accommodate a wide variety of requirements, such as launching Spot and On-Demand Instances.

☒ **Adhere to launch template**

The launch template determines the purchase option (On-Demand or Spot) and instance type.

☐ **Combine purchase options and instance types**

Specify how much On-Demand and Spot capacity to launch and multiple instance types (optional). This choice is most helpful for optimizing the scale and cost for a fleet of instances.

Network [Info](#)

For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones. The default VPC and default subnets are suitable for getting started quickly.

VPC

vpc-0b978358e22761686 (my-lab-vpc)
10.0.0.0/16

5) Select VPC

[Create a VPC](#)

Subnets

Select subnets

us-east-1a | subnet-0ef43ef1cfcb561a0 (lab-sn-public-1A)
10.0.0.0/24

6) Select Some Subnets

us-east-1b | subnet-03b7f8298553c4646 (lab-sn-public-1B)
10.0.2.0/24

7) Click Next

[Create a subnet](#)

Cancel

Previous

Skip to review

Next

Configure advanced options [Info](#)

Choose a load balancer to distribute incoming traffic for your application across instances to make it more reliable and easily scalable. You can also set options that give you more control over health check replacements and monitoring.

Load balancing - *optional* [Info](#)

8) Select Attach to Existing Load Balancer

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.

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

my-lab-target | HTTP
Application Load Balancer: my-lab-alb

9) Select Your Load Balancer Target Group

Health checks - *optional*

Health check type [Info](#)

EC2 Auto Scaling automatically replaces instances that fail health checks. If you enabled load balancing, you can enable ELB health checks in addition to the EC2 health checks that are always enabled.

☒ EC2 ☐ ELB

Health check grace period

The amount of time until EC2 Auto Scaling performs the first health check on new instances after they are put into service.

300 seconds

Additional settings - *optional*

Monitoring [Info](#)

☐ Enable group metrics collection within CloudWatch

10) Click Next

Cancel

Previous

Skip to review

Next

Health checks

Health checks increase availability by replacing unhealthy instances. When you use multiple health checks, all are evaluated, and if at least one fails, instance replacement occurs.

EC2 health checks

[Always enabled](#)

Additional health check types - optional [Info](#)

☒ **Turn on Elastic Load Balancing health checks** [Recommendation](#)

Elastic Load Balancing monitors whether instances are available to handle requests. When it reports an unhealthy instance, EC2 Auto Scaling can replace it on its next periodic check.

[EC2 Auto Scaling will start to detect and act on health checks performed by Elastic Load Balancing. To avoid unexpected terminations, first verify the settings of these health checks in the Load Balancer console.](#)

[View or modify your health checks](#)

VPC Lattice can monitor whether instances are available to handle requests. If it considers a target as failed a health check, EC2 Auto Scaling replaces it after its next periodic check.

Health check grace period [Info](#)

This time period delays the first health check until your instances finish initializing. It doesn't prevent an instance from terminating when placed into a non-running state.

seconds

Additional settings

Monitoring [Info](#)

☐ Enable group metrics collection within CloudWatch

Default instance warmup [Info](#)

The amount of time that CloudWatch metrics for new instances do not contribute to the group's aggregated instance metrics, as their usage data is not reliable yet.

☐ Enable default instance warmup

Cancel Skip to review Previous **Next**

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

Minimum capacity

Maximum capacity

11) Set Desired, Min, and Max Capacity

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

Metric type

Target value

12) Set Target Tracking for CPU Utilization

Instances need
 seconds warm up before including in metric

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

Instance scale-in protection - optional

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

☐ Enable instance scale-in protection

13) Click Next

Cancel Previous Skip to review **Next**

4: Finally create the ASG.

Verify and Test the ALB

View the Health check in your Target Group Details. Both instances should be healthy.

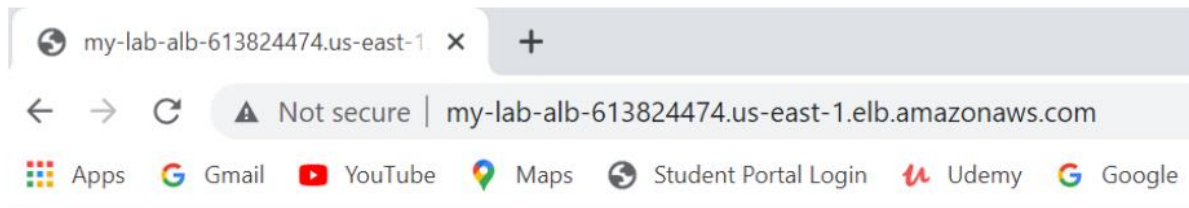
The screenshot shows the AWS Management Console interface. On the left is a navigation menu with categories like EC2 Dashboard, Instances, Images, Elastic Block Store, and Network & Security. The main content area is titled 'Details' for a target group. It shows metadata like 'Target type: Instance', 'Protocol: Port: HTTP: 80', and 'VPC: vpc-005c2c5c828aafe34'. Below this is a summary table for target health:

Total targets	Healthy	Unhealthy	Unused	Initial	Draining
2	2	0	0	0	0

Below the summary table is a section 'Distribution of targets by Availability Zone (AZ)' with a note to select values for filters. At the bottom, the 'Targets' tab is active, showing a table of 'Registered targets (2)'. The table has columns for Instance ID, Name, Port, Zone, Health status, and Health status details. Both targets are listed as 'healthy'.

Instance ID	Name	Port	Zone	Health status	Health status details
i-0b9fbe760e45ae4c4	lab4-my-ec2-first-ins...	80	us-east-1a	healthy	
i-02dc3b774ae789f85	lab4-my-ec2-second ...	80	us-east-1b	healthy	

Test DNS with Web Browser



Hello from my EC2 Instance in Autoscaling Group Behind an ALB

2: You can use EC2 stress tool to test out the scaling out.