

## Assignment 4 – ALB and ASG – Step by step

### Create Security Groups

- 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 <input type="checkbox"/> my-lab-alb-sg	Security group ID <input type="checkbox"/> sg-03e5e025e377518eb	Description <input type="checkbox"/> Lab Application Load Balancer Security Group	VPC ID <input type="checkbox"/> vpc-0b978358e22761686 <input checked="" type="checkbox"/>
Owner <input type="checkbox"/> 409673912482	Inbound rules count 1 Permission entry	Outbound rules count 1 Permission entry	

**Inbound rules    Outbound rules    Tags**

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 <input type="checkbox"/> my-lab-EC2-Server-sg	Security group ID <input type="checkbox"/> sg-0a370c15c5b405b61	Description <input type="checkbox"/> Web Server Security Group	VPC ID <input type="checkbox"/> vpc-0b978358e22761686 <input checked="" type="checkbox"/>
Owner <input type="checkbox"/> 409673912482	Inbound rules count 1 Permission entry	Outbound rules count 1 Permission entry	

**Inbound rules    Outbound rules    Tags**

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

my-lab-alb-sg  
Security Group

## Create an ALB

Go to the Load Balancers Display from the EC2 Dashboard

The screenshot shows the AWS EC2 Dashboard. On the left, there is a navigation sidebar with sections for Snapshots, Lifecycle Manager, Network & Security (Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces), Load Balancing (Load Balancers, Target Groups), Auto Scaling (Launch Configurations, Auto Scaling Groups), and three ellipsis (...). The Load Balancing section is expanded, and the Load Balancers item is highlighted with a red arrow pointing to it. The main area is titled "Resources" and displays the following information: Instances (running) 0, Dedicated Hosts 0, Elastic IPs 0, Instances 0, Key pairs 4, Load balancers 0, Placement groups 0, Security groups 7, Snapshots 0, and Volumes 0. A blue box highlights the "Load balancers" row, and a red arrow points from the sidebar to this box. Below the resources table, there is a message about deploying Microsoft SQL Server Always On availability groups.

The screenshot shows the "Load Balancers" list page. At the top, there is a "Create Load Balancer" button and an "Actions" dropdown menu. A red arrow points from the "Create Load Balancer" button to a blue box containing the text "2) Click on Create Load Balancer". Below the button is a search bar with the placeholder "Filter by tags and attributes or search by keyword". Underneath the search bar are filters for Name, DNS name, State, VPC ID, and Availability Zones. A message at the bottom states "You do not have any load balancers in this region."

### Select load balancer type

Elastic Load Balancing supports four types of load balancers: Application Load Balancers, Network Load Balancers, Gateway Load Balancers, and Classic Load Balancers. Choose the load balancer type that meets your needs.

[Learn more about which load balancer is right for you](#)

The screenshot shows the "Select load balancer type" page. It features three cards: "Application Load Balancer" (HTTP, HTTPS, Create), "Network Load Balancer" (TCP, TLS, UDP, Create), and "Gateway Load Balancer" (IP, Create). Each card has a brief description below it. A red arrow points from the "Create" button in the Application Load Balancer card to a blue box containing the text "3) Click on Create Application Load Balancer".

Application Load Balancer	Network Load Balancer	Gateway Load Balancer
HTTP HTTPS <a href="#">Create</a>	TCP TLS UDP <a href="#">Create</a>	IP <a href="#">Create</a>
Choose an Application Load Balancer when you need a flexible feature set for your web 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.	Choose a Network Load Balancer when you need ultra-low latency, high-throughput, and high-availability connectivity. Network Load Balancers are capable of handling millions of requests per second securely.	Choose a Gateway Load Balancer when you need to manage a fleet of third-party virtual appliances that support GENEVE. These appliances improve security, compliance, and policy enforcement.

## Step 1: Configure Load Balancer

### Basic Configuration

To configure your load balancer, provide a name, select a scheme, specify one or more listeners, and select a network. The default configuration is an Internet-facing load balancer in the selected network with a listener that receives HTTP traffic on port 80.

Name	<input type="text" value="my-lab-alb"/>	4) Name Load Balancer
Scheme	<input checked="" type="radio"/> internet-facing <input type="radio"/> internal	
IP address type	<input type="text" value="ipv4"/>	

### Listeners

A listener is a process that checks for connection requests, using the protocol and port that you configured.

Load Balancer Protocol	Load Balancer Port
HTTP	80

Cancel    Next: Configure Security Settings

### Availability Zones

Specify the Availability Zones to enable for your load balancer. The load balancer routes traffic to the targets in these Availability Zones only. You can specify only one subnet per Availability Zone. You must specify subnets from at least two Availability Zones to increase the availability of your load balancer.

VPC	<input type="text" value="vpc-0b978358e22761686 (10.0.0.0/16)   my-lab-vpc"/>	5) Select VPC
Availability Zones	<input checked="" type="checkbox"/> us-east-1a <input type="text" value="subnet-0ef43ef1fcfb561a0 (lab-sn-public-1A)"/> <input checked="" type="checkbox"/> us-east-1b <input type="text" value="subnet-03b7f8298553c4646 (lab-sn-public-1B)"/>	6) Select At Least 2 AZ Zones and Subnets
	IPv4 address    Assigned by AWS	
	IPv4 address    Assigned by AWS	
	<input type="checkbox"/> us-east-1c <input type="text" value="subnet-028d2f8b4b0b12258 (lab-sn-public-1c)"/>	

Additional AWS services can be integrated with this load balancer at launch when you enable them below. 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**  Create an accelerator to get static IP addresses and improve the performance and availability of your application. [Learn more](#)  
Additional charges apply

Your Accelerator will be created with the following name that you can customize. Once your Accelerator is created you can manage it from the Global Accelerator console.

Accelerator name  
  
Maximum 64 characters. Letters and numbers only.

### Tags

7) Click Next

Cancel    Next: Configure Security Settings

...

## Step 2: Configure Security Settings

**⚠ Improve your load balancer's security.** Your load balancer is not using any secure listener.  
If your traffic to the load balancer needs to be secure, use the HTTPS protocol for your front-end connection. You can go back to the first step to add/configure secure listeners under [Basic Configuration](#) section. You can also continue with current settings.

8) Click Next

[Cancel](#) [Previous](#) [Next: Configure Security Groups](#)

## Step 3: Configure Security Groups

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

Assign a security group     Create a new security group  
 Select an existing security group

Filter [VPC security groups](#)

Security Group ID	Name	Description	Actions
sg-002d4b487ca1292d2	default	default VPC security group	<a href="#">Copy to new</a>
sg-0d09b0bf676ce516f	launch-wizard-2	launch-wizard-2 created 2021-07-08T19:37:07.572-05:00	<a href="#">Copy to new</a>
sg-0fc356187933ae278	launch-wizard-3	launch-wizard-3 created 2021-07-08T20:58:44.588-05:00	<a href="#">Copy to new</a>
<input checked="" type="checkbox"/> sg-03e5e025e377518eb	my-lab-alb-sg	Lab Application Load Balancer Security Group	<a href="#">Copy to new</a>
<input type="checkbox"/> sg-0a370c15c5b405b61	my-lab-EC2-Server-sg	Web Server Security Group	<a href="#">Copy to new</a>

7) Select the ALB Security Group you Created

9) Click Next

[Cancel](#) [Previous](#) [Next: Configure Routing](#)

## Step 4: Configure Routing

Your load balancer routes requests to the targets in this target group using the protocol and port that you specify here. It also performs health checks on the targets using these settings. The target group you specify in this step will apply to all of the listeners configured on this load balancer. You can edit or add listeners after the load balancer is created.

### Target group

Target group	<input type="text" value="New target group"/>
Name	<input type="text" value="my-lab-target"/>
Target type	<input checked="" type="radio"/> Instance <input type="radio"/> IP <input type="radio"/> Lambda function
Protocol	<input type="text" value="HTTP"/>
Port	<input type="text" value="80"/>
Protocol version	<input checked="" type="radio"/> HTTP1 Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.

10) Name Target Group

11) Select Instance

Health checks

Protocol: HTTP  
Path: /

Advanced health check settings

**12) Click Next**

Cancel Previous **Next: Register Targets**

Step 5: Register Targets

Register targets with your target group. If you register a target in an enabled Availability Zone, the load balancer starts routing requests to the targets as soon as the registration process completes and the target passes the initial health checks.

**Registered targets**

To deregister instances, select one or more registered instances and then click Remove.

Remove	Instance	Name	Port	State	Security groups	Zone
No instances available.						

**Instances**

To register additional instances, select one or more running instances, specify a port, and then click Add. The default port is the port specified for the target group. If the instance is already registered on the specified port, you must specify a different port.

**13) Click Next**

Add to registered on port 80

Cancel Previous **Next: Review**

Step 6: Review

Please review the load balancer details before continuing

**Load balancer**

- Name: my-lab-alb
- Scheme: internet-facing
- Listeners: Port:80 - Protocol:HTTP
- IP address type: ipv4
- VPC: vpc-0b978358e22761686 (my-lab-vpc)
- Subnets: subnet-0ef43ef1cfcb561a0 (lab-sn-public-1A), subnet-03b7f8298553c4646 (lab-sn-public-1B)
- Tags

**Security groups**

- Security groups: sg-03e5e025e377518eb

**Routing**

- Target group: New target group
- Target group name: my-lab-target

**14) Click Create**

Cancel Previous **Create**

## Listener rules

# Practice Listener Rules and Lambda as target

## Part-1: Creating Lambda Function

1.a Search for **Lambda** and open it.

The screenshot shows the AWS Management Console search results for the term 'Lambda'. The search bar at the top contains 'Lambda'. The results are categorized under 'Services' and 'Features'. The 'Lambda' service card is highlighted with a red box. It includes the text 'Run code without thinking about servers'. Below it are cards for 'CodeBuild', 'AWS Signer', and 'Amazon Inspector'. In the 'Features' section, cards for 'Local processing', 'Target groups', and 'Publish applications' are listed. The right side of the screen shows a sidebar with 'AWS' and various informational links.

1.b In the navigation pane, choose **Functions** and then choose **Create function**.

The screenshot shows the AWS Lambda Functions page. The left navigation pane has 'Functions' selected, which is highlighted with a red box. The main area displays a table of existing Lambda functions, and the 'Create function' button is highlighted with a red box. The table columns include Function name, Description, Package type, Runtime, and Last modified. The functions listed are: RedshiftOverwatch, RoleCreationFunction, RedshiftEventSubscription, PutItemLambda, MyFirstLambda, and MainMonitoringFunction.

1.c Fill out the **Function name** and click on **Change default execution role**.

**Basic Information**

Function name  
Enter a name that describes the purpose of your function.  
**App1**

Runtime Info  
Choose the language to use for your function. Note that the console code editor supports only Node.js, Python, and Ruby.  
**Node.js 18.x**

Architecture Info  
Choose the instruction set architecture you want for your function code.  
**x86\_64**

Permissions Info  
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▶ Change default execution role

▶ Advanced settings

Cancel **Create function**

1.d Choose “Use and Existing Role” and select **LabRole** from the dropdown menu of Existing role.

Runtime Info  
Choose the language to use for your function. Note that the console code editor supports only Node.js, Python, and Ruby.  
**Node.js 18.x**

Architecture Info  
Choose the instruction set architecture you want for your function code.  
**x86\_64**

Permissions Info  
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▼ Change default execution role

Execution role  
Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console [\[ \]](#).  
 Create a new role with basic Lambda permissions  
 Use an existing role  
 Create a new role from AWS policy templates

Existing role  
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.  
**LabRole**

View the LabRole role [\[ \]](#) on the IAM console.

▼ Advanced settings

Enable Code signing [\[ \]](#)

1.e Click on Advance Settings.

- a. Select **Enable Function URL**
- b. On Auth Type Select **None**
- c. Select **Configure cross-origin resource sharing (CORS)**
- d. Click on **Create Function**

The screenshot shows the AWS Lambda function creation interface. In the 'Advanced settings' section, the 'Enable function URL' checkbox is checked. Under 'Auth type', 'NONE' is selected. In the 'Function URL permissions' section, the 'Configure cross-origin resource sharing (CORS)' checkbox is checked. The 'Create function' button is highlighted with a red box.

1.f Add the header in function as given below, so browser won't download a file when calling the function through Alb. Click on Deploy button.

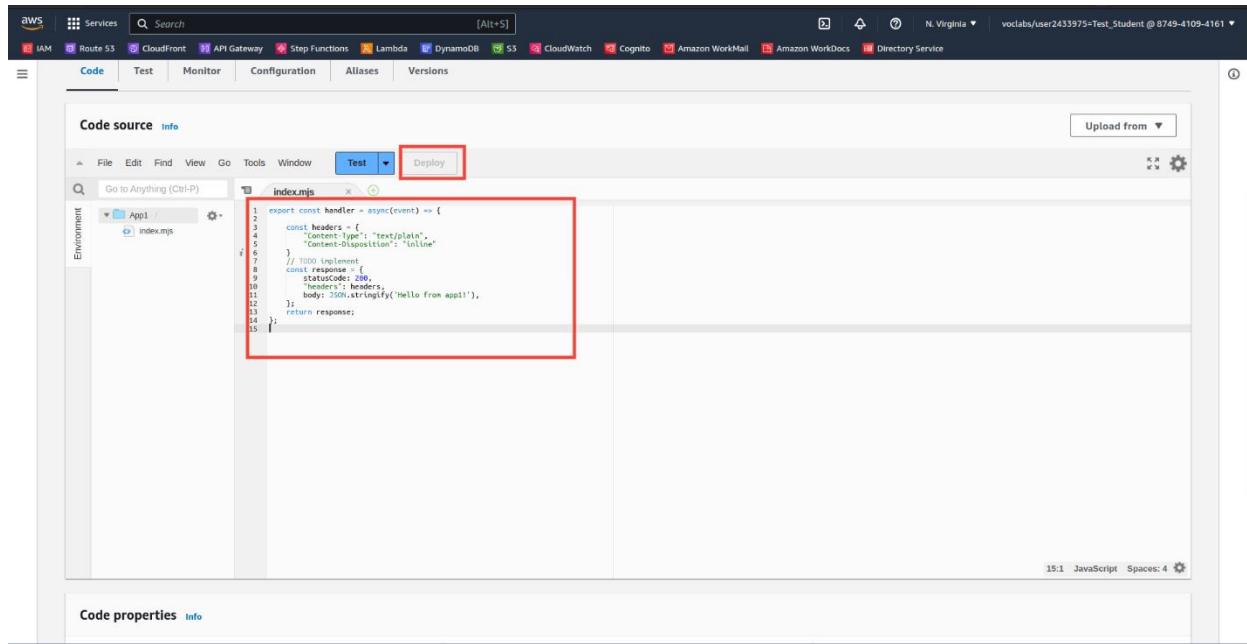
```
export const handler = async(event) => {
```

```
    const headers = {
        "Content-Type": "text/plain",
        "Content-Disposition": "inline"
    }
```

```
// TODO implement
```

```
    const response = {
        statusCode: 200,
        "headers": headers,
        body: JSON.stringify('Hello from app1!'),
    };
    return response;
}
```

};



The screenshot shows the AWS Lambda console interface. The top navigation bar includes services like IAM, Route 53, CloudFront, API Gateway, Step Functions, Lambda, DynamoDB, CloudWatch, Cognito, Amazon WorkDocs, Amazon WorkMail, and Directory Service. The user is signed in as vocabs/user2433975+Test\_Student @ 8749-4109-4161. Below the navigation, there are tabs for Code, Test, Monitor, Configuration, Aliases, and Versions. The Code tab is selected. The main area displays the code editor for the 'index.js' file under the 'App1' function. The code is as follows:

```
1 export const handler = async(event) => {
2   const headers = [
3     "Content-Type": "text/plain",
4     "Content-Disposition": "inline"
5   ]
6   // TODO implement
7   const response = {
8     statusCode: 200,
9     headers: headers,
10    body: JSON.stringify('Hello from app1!'),
11  };
12  return response;
13}
14};
```

A red box highlights the entire code block. At the bottom right of the code editor, it says '15:1 JavaScript Spaces: 4'. Below the code editor, there are tabs for Code properties and Info.

**Create another Lambda with a different name. Follow the same steps to create Lambda.**

## Part-2: Create Security Group for ALB.

2. a Go to EC2 Console.

Select Security Groups from Side Bar and Click on Create Security Group.

The screenshot shows the AWS Management Console with the VPC service selected. In the left navigation pane, 'Security Groups' is highlighted with a red box. The main content area displays a table of 11 security groups. The columns include Name, Security group ID, Security group name, VPC ID, Description, Owner, and Inbound rules count. A red box highlights the 'Create security group' button at the top right of the table.

2. b Fill out Security group name, Description and Select VPC. In Inbound Rules section Select type as Http and 0.0.0.0/0 as CIDR block.

The screenshot shows the 'Create security group' wizard. The first step, 'Basic details', has the 'Security group name' field set to 'Alb-Lambda-sg'. The 'Description' field is also filled with 'Alb-Lambda-sg'. Under 'VPC', the dropdown shows 'vpc-0f5cdad7232fa1fda'. The second step, 'Inbound rules', shows a single rule: 'Type: HTTP', 'Protocol: TCP', 'Port range: 80', 'Source: Anywhere', and 'CIDR block: 0.0.0.0/0'. The 'Add rule' button is visible at the bottom left of this section.

2. c Click on Create Security Group.

Inbound rules

Type: HTTP, Protocol: TCP, Port range: 80, Source: Anywhere..., Description: 0.0.0.0/0

Outbound rules

Type: All traffic, Protocol: All, Port range: All, Destination: Custom, Description: 0.0.0.0/0

Tags - optional

No tags associated with the resource.

Add new tag

Add rule

Create security group

## Part-3: Create Target Group TG1 and TG2.

3. a Select Security Groups from Side Bar and Click on Create Security Group.

Successfully created target group: TG1

EC2 > Target groups

Target groups (6) Info

Name	ARN	Port	Protocol	Target type	Load balancer	VPC ID
Alb-tg-1	arn:aws:elasticloadbalancing:us-east-1:123456789012:targetgroup/Alb-tg-1/1234567890123456	-	-	Lambda	ALB-Lambda	-
Alb-tg-2	arn:aws:elasticloadbalancing:us-east-1:123456789012:targetgroup/Alb-tg-2/1234567890123456	-	-	Lambda	ALB-Lambda	-
TG1	arn:aws:elasticloadbalancing:us-east-1:123456789012:targetgroup/TG1/1234567890123456	-	-	Lambda	None associated	-
TgDemo	arn:aws:elasticloadbalancing:us-east-1:123456789012:targetgroup/TgDemo/1234567890123456	80	HTTP	Instance	AlbDemo	vpc-0f5cdad7232fa1fda

0 target groups selected

Select a target group above.

Actions

Create target group

Target Groups

3. b Select Lambda function and fill out the Target group name.

Screenshot of the AWS Lambda function configuration page for Step 1: Specify group details. The 'Lambda function' option is selected and highlighted with a red box. The target group name 'TG1' is also highlighted with a red box.

### 3.c Click Next.

Screenshot of the AWS Lambda function configuration page for Step 2: Register targets. The 'Next' button is highlighted with a red box.

### 3.d Now Select the Function and Click on Create Target Group.

The screenshot shows the 'Register targets' step in the 'Create target group' wizard. It's titled 'Lambda function' and asks for a single Lambda function as the target. A dropdown menu is open, showing 'App1' selected. Below it are options for 'Version' (selected) and 'Alias', and two other radio button options: 'Enter a Lambda function ARN.' and 'Add a function later'. At the bottom right are 'Cancel', 'Previous', and 'Create target group' buttons.

**Create another Target Group with a different name. Follow the same steps to create Target Group.**

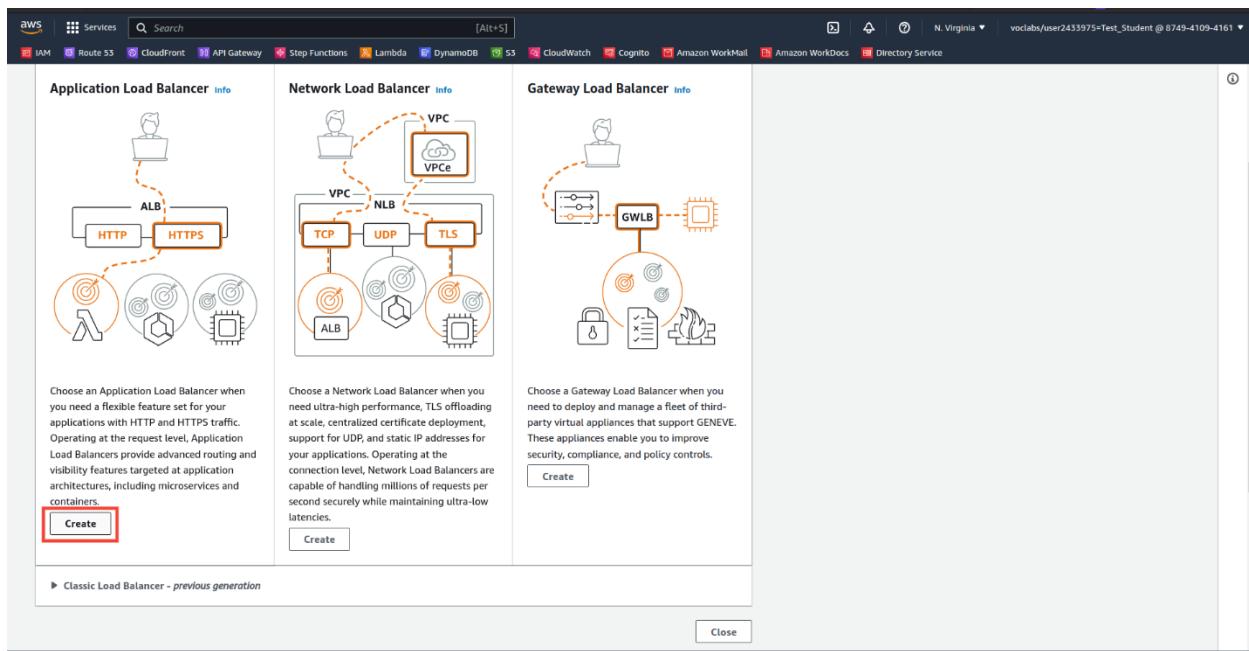
## Part-4: Create Application Load Balancer.

4. a Select Load Balancers from the sidebar and Click on Create load balancer.

The screenshot shows the 'Load balancers' section of the EC2 dashboard. It lists three existing load balancers: 'my-nlb', 'my-alb', and 'AlbDemo'. The 'Create load balancer' button is highlighted with a red box. On the left sidebar, the 'Load Balancing' section is expanded, with 'Load Balancers' also highlighted with a red box.

Name	DNS name	State	VPC ID	Availability Zones	Type	Date created
my-nlb	my-nlb-f327d7fce235b24...	Active	vpc-0f5cdad7232fa1fda	2 Availability Zones	network	March 2, 2023, 13:44 (UTC-06:00)
my-alb	my-alb-355457061.us-eas...	Active	vpc-0f5cdad7232fa1fda	2 Availability Zones	application	March 2, 2023, 10:58 (UTC-06:00)
AlbDemo	AlbDemo-960472562.us-e...	Active	vpc-0f5cdad7232fa1fda	2 Availability Zones	application	March 3, 2023, 14:03 (UTC-06:00)

4. b Select **Create** from Application Load Balancer.



#### 4. c Fill out the Load balancer name.

The screenshot shows the 'Create Application Load Balancer' wizard. Step 1: Basic configuration. The 'Load balancer name' field is filled with 'ALB-Lambda'. A red box highlights this input field.

**Basic configuration**

**Load balancer name**  
Name must be unique within your AWS account and cannot be changed after the load balancer is created.  
**ALB-Lambda**

A maximum of 52 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

**Scheme** Info  
Scheme cannot be changed after the load balancer is created.  
 **Internet-facing**  
An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. [Learn more](#) ?  
 **Internal**  
An internal load balancer routes requests from clients to targets using private IP addresses.

**IP address type** Info  
Select the type of IP addresses that your subnets use.  
 **IPv4**  
Recommended for internal load balancers.  
 **Dualstack**  
Includes IPv4 and IPv6 addresses.

**Network mapping** Info

#### 4. d Select Two availability zones. For this demo we will select us-east-1a and us-east-1b.

**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. Only VPCs with an internet gateway are enabled for selection. The selected VPC cannot be changed after the load balancer is created. To confirm the VPC for your targets, view your [target groups](#).

vpc-0f5cdad7232fa1fda  
IPv4: 172.51.0.0/16

**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-az2)

Subnet  
subnet-03e3088a6bc6a72d2

IPv4 settings  
Assigned by AWS

us-east-1b (use1-az4)

Subnet  
subnet-032376d9ec307c994

IPv4 settings  
Assigned by AWS

4. e Now we need to select the Security Group. Select the Security Group that we created in the previous step.

**Security groups** [Info](#)

A security group is a set of firewall rules that control the traffic to your load balancer.

**Security groups**

Select up to 5 security groups  
 Alb-Lambda-sg sg-0a829b05fe3a8ac86 X  
Create new security group

**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

Protocol	Port	Default action	Info
HTTP	: 80 1-65535	Forward to Select a target group	<a href="#">Create target group</a>

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.

4. f Now under listeners and routing, select TG1 as the target group. We created this target group in the previous step.

The screenshot shows the AWS Lambda console with the 'Security groups' section open. A new security group, 'Alb-Lambda-sg', has been created and is listed. The 'Listeners and routing' section is also visible, showing a listener for port 80 forwarding to target group TG1.

#### 4. g Click on Create load balancer.

The screenshot shows the AWS Load Balancers console with the 'Tags - optional' section and 'Summary' section visible. The 'Create load balancer' button is highlighted with a red box.

#### 4. h Now Go to Load Balancers and Click on the load Balancer that you created.

The screenshot shows the AWS CloudFront Load Balancers page. The left sidebar includes sections for IAM, Route 53, CloudFront, API Gateway, Step Functions, Lambda, DynamoDB, S3, CloudWatch, Cognito, Amazon WorkMail, Amazon WorkDocs, and Directory Service. Under the 'Load Balancing' section, 'Load Balancers' is selected. The main content area displays a table titled 'Load balancers (4)'. The table columns are: Name, DNS name, State, VPC ID, Availability Zones, Type, Date created, and Instance. The rows show the following details:

Name	DNS name	State	VPC ID	Availability Zones	Type	Date created	Instance
my-nlb	my-nlb-f527d7fce235b24...	Active	vpc-0f5cdad7232fa1fda	2 Availability Zones	network	March 2, 2023, 13:44 (UTC-06:00)	-
my-alb	my-alb-355457061.us-eas...	Active	vpc-0f5cdad7232fa1fda	2 Availability Zones	application	March 2, 2023, 10:58 (UTC-06:00)	-
AlbDemo	AlbDemo-960472562.us-e...	Active	vpc-0f5cdad7232fa1fda	2 Availability Zones	application	March 3, 2023, 14:03 (UTC-06:00)	-
<b>ALB-Lambda</b>	<b>ALB-Lambda-1678675122...</b>	<b>Active</b>	<b>vpc-0f5cdad7232fa1fda</b>	<b>2 Availability Zones</b>	<b>application</b>	<b>March 6, 2023, 21:26 (UTC-06:00)</b>	<b>-</b>

At the bottom of the table, it says '0 load balancers selected'.

4.i Scroll down and select the protocol. After that Click on Actions and select Manage rules.

The screenshot shows the AWS CloudFront Listener configuration for the ALB-Lambda load balancer. The left sidebar includes sections for IAM, Route 53, CloudFront, API Gateway, Step Functions, Lambda, DynamoDB, S3, CloudWatch, Cognito, Amazon WorkMail, Amazon WorkDocs, and Directory Service. Under the 'Load Balancing' section, 'Load Balancers' is selected. The main content area shows the ALB-Lambda load balancer details. The 'Listeners' tab is selected, showing one listener for port 80. The 'Manage rules' option in the Actions menu is highlighted with a red box.

Protocol	Port	ARN	Security policy	Default SSL cert	Default routing rule
HTTP	80	Not applicable	Not applicable	Not applicable	1. Forward to o T1: 1 (100%) o Group-level stickiness: Off

The Actions menu includes options: View listener details, Edit listener, Add SSL certificates for SNI, Manage rules (highlighted with a red box), Manage tags, and Delete listener.

4. j Click on Edit Icon.

The screenshot shows the AWS Lambda console with the URL [ALB-Lambda | HTTP:80](#). A red box highlights the edit icon (pencil) in the top navigation bar. The page displays a single rule: "HTTP 80: default action". The "IF" condition is checked for "Requests otherwise not routed". The "THEN" section shows "Forward to..." with "TG1: 1 (100%)". A note states "Group-level stickiness: Off". Below the rule table, there is a note: "Select the rule to edit. Each rule must include one action of type forward, redirect, fixed response." A "Rule limits for condition values, wildcards, and total rules." link is also present.

#### 4. k Delete the THEN section and Add Return Fixed Response Action.

The screenshot shows the AWS Lambda console with the URL [ALB-Lambda | HTTP:80](#). A red box highlights the edit icon (pencil) in the top navigation bar. The page displays a single rule: "HTTP 80: default action". The "IF" condition is checked for "Requests otherwise not routed". The "THEN" section has been removed, and a new "THEN" section has been added with the "Return fixed response" action selected. A note at the bottom states: "Note: Additional actions are available for HTTPS listeners." A "Cancel" and "Update" button are visible at the bottom right.

4.l

Change Response Code to 200 and Change Response Body to "Fixed Response". Lastly Click Update.

The screenshot shows the AWS Lambda function configuration page. A specific rule is being edited for the ALB-Lambda target group. The 'THEN' section of the rule configuration is displayed, showing a fixed response with a status code of 200 and a body of 'Fixed Response'. The 'Update' button at the top right of the dialog is highlighted with a red box.

4.n Now click first Click on  $\oplus$ . Then click on Insert Rule.

The screenshot shows the AWS Lambda function configuration page. A specific rule is being edited for the ALB-Lambda target group. The 'IF' section of the rule configuration is displayed, showing a condition 'Requests otherwise not routed'. The '+ Insert Rule' button at the top right of the dialog is highlighted with a red box.

4.o Now click on Add condition and Select Path.

The screenshot shows the AWS Lambda function configuration interface. Under the 'ALB-Lambda | HTTP:80' tab, there are two rules listed. The first rule, '1' (the default), has its 'IF (all match)' condition set to 'Path...' (which is highlighted with a red box). The 'THEN' action for this rule is 'Return fixed response 200 (more...)'. The second rule, 'last', is a 'HTTP 80: default action' with the condition 'Requests otherwise not routed' and the 'THEN' action 'Return fixed response 200 (more...)'. A note at the bottom left states: 'A rule ID (ARN) is generated when you save your rule. This rule cannot be moved or deleted.'

4.p Write path as /app1. Click on Add action and select Forward to.

The screenshot shows the AWS Lambda function configuration interface. Under the 'ALB-Lambda | HTTP:80' tab, there are two rules listed. The first rule, '1', has its 'IF (all match)' condition set to 'Path...' with the value '/app1' (highlighted with a red box). The 'THEN' action for this rule is 'Forward to...' (highlighted with a red box). The second rule, 'last', is a 'HTTP 80: default action' with the condition 'Requests otherwise not routed' and the 'THEN' action 'Return fixed response 200 (more...)'. A note at the bottom left states: 'A rule ID (ARN) is generated when you save your rule. This rule cannot be moved or deleted.'

4 q. Now Select Target Group as TG1 and Click on Save.

Screenshot of the AWS CloudFront Rules configuration page for ALB-Lambda | HTTP:80. A new rule is being created with the following details:

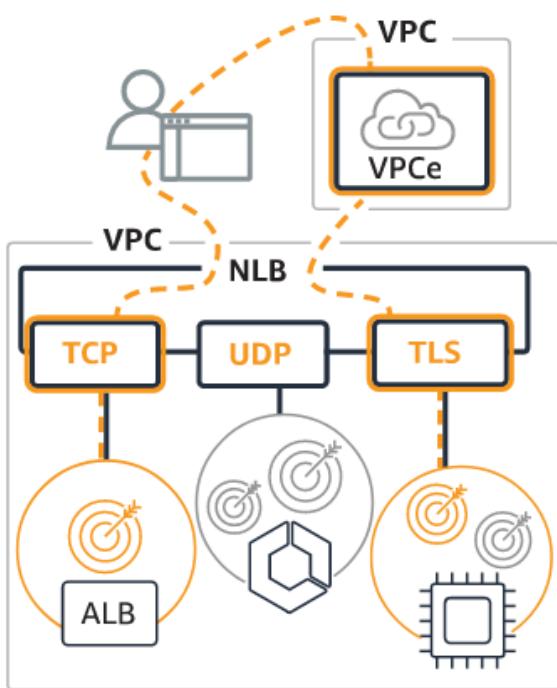
- RULE ID:** 1
- Path...**: `/app1`
- THEN:**
  - Forward to...**: Target group : Weight (0-999) - TG1 (Traffic distribution 100%)
  - Group-level stickiness**: checked
  - Add action**: Return fixed response 200 (more...)
- Last Action:** HTTP 80: default action (Requests otherwise not routed)

The "Save" button is highlighted in red at the top right.

**Create Rule for TG2. Add path as /app2 and target as TG2.**

Create an NLB

## Network Load Balancer Info



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 latencies.

[Create](#)

a. Spin up 2 instances with different HTML content in us-east-1a, us-east-1b AZs.

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input type="checkbox"/>	-	i-0452e560715ee832a	<span>Running</span>	t2.micro	<span>2/2 checks passed</span>	No alarms	us-east-1a
<input checked="" type="checkbox"/>	myNLbE1	i-03f9925915e9136a6	<span>Running</span>	t2.micro	<span>Initializing</span>	No alarms	us-east-1a
<input checked="" type="checkbox"/>	myNLBE2	i-009a5fdf725c0d8d2	<span>Running</span>	t2.micro	-	No alarms	us-east-1b

b. Add the instances in us-east-1a, us-east-1b to the target group of the NLB.

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

Protocol      Port  
 :

VPC  
Select the VPC with the instances that you want to include in the target group.

## Register targets

This is an optional step to create a target group. However, to ensure that your load balancer routes traffic to this target group you must register your targets.

Available instances (2/3)

	Instance ID	Name	State	Security groups	Zone	Subnet ID
<input type="checkbox"/>	i-0452e560715ee832a		<span>running</span>	MyAppBehindAlb	us-east-1a	subnet-0b5aeb0697c77d6a5
<input checked="" type="checkbox"/>	i-03f9925915e9136a6	myNLbE1	<span>running</span>	default	us-east-1a	subnet-0b5aeb0697c77d6a5
<input checked="" type="checkbox"/>	i-009a5fdf725c0d8d2	myNLBE2	<span>running</span>	default	us-east-1b	subnet-0d2438927d9c21121

c. Update the target group and deselect Preserve client IP addresses

## Edit attributes

**Attributes** Restore defaults

**Deregistration delay**  
The time to wait for in-flight requests to complete while deregistering a target. During this time, the state of the target is draining.

seconds  
0-3600

**Connection termination on deregistration — recommended**  
If enabled, your Network Load Balancer will terminate active connections when deregistration delay is reached.

**Stickiness**  
The type of stickiness associated with this target group. If enabled, the load balancer binds a client's session to a specific instance within the target group.

**Proxy protocol v2**  
Before you enable proxy protocol v2, make sure that your application targets can process proxy protocol headers otherwise your application might break.

**Preserve client IP addresses**  
Preserve client IP addresses and ports in the packets forwarded to targets.

f. Grab private subnets. Update the instance's security group to allow access from the NLB nodes created in those subnets.

Listeners Network mapping Monitoring Integrations Attributes Tags

**Network mapping** Targets in the listed zones and subnets are available for traffic from the load balancer using the IP addresses shown. Edit IP address type Edit subnets

VPC <a href="#">vpc-07c5c367badcf2e8d</a> IPv4: 172.31.0.0/16 IPv6 :-	IP address type IPv4
--	-------------------------

**Mappings** Targets in the listed zones and subnets are available for traffic from the load balancer using the IP addresses shown.

▼	IPv4 address	Private IPv4 address	▼	IPv6 address
<a href="#">5103ff283a1db</a>	Assigned by AWS	Assigned from CIDR 172.31.80.0/20		Not Applicable
<a href="#">db9fbfb79edff4</a>	Assigned by AWS	Assigned from CIDR 172.31.16.0/20		Not Applicable

**Inbound rules (4)**

Filter security group rules

G Manage tags Edit inbound rules

< 1 > ⚙

Rule name	IP version	Type	Protocol	Port range	Source	Description
Odca9daad	IPv4	HTTP	TCP	80	172.31.80.0/20	-
b637663cf	-	SSH	TCP	22	sg-06ea6b051e1d354...	-
a7002a56	-	HTTP	TCP	80	sg-06ea6b051e1d354...	-
7f318d144	IPv4	HTTP	TCP	80	172.31.16.0/20	-

## Create a launch template

[Go to Launch Templates Display from EC2 Display](#)

New EC2 Experience [Learn more](#)

**EC2 Dashboard**

- Events
- Tags
- Limits
- Instances**
  - Instances [New](#)
  - Instance Types
  - Launch Templates** 1) Click Launch Templates
  - Spot Requests
  - Savings Plans
  - Reserved Instances [New](#)
  - Dedicated Hosts
  - Scheduled Instances
  - Capacity Reservations

**Resources**

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

Instances (running)	0	Dedicated Hosts	0
Elastic IPs	0	Instances	0
Key pairs	4	Load balancers	0
Placement groups	0	Security groups	9
Snapshots	0	Volumes	0

## Create Launch Template

EC2 > Launch templates

**Launch templates (1) [Info](#)**

[Actions ▾](#) **Create launch template**

Filter by tags or properties or search by keyword

Launch template ID	Launch template name	Default version
lt-0add0ae0ee0d310d3		

1) Click 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

2) Name Template

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

Selecting Guidance will show more Details

► 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.

#### ▼ Amazon machine image (AMI) - required [Info](#)

AMI - *required*

Amazon Linux 2 AMI (HVM), SSD Volume Type

ami-0dc2d3e4c0f9ebd18

Catalog: Quick Start virtualization: hvm architecture: 64-bit (x86)

3) Select AMI

#### ▼ Instance type [Info](#)

Instance type

t2.micro

Family: t2 1 vCPU 1 GiB Memory

On-Demand Linux pricing: 0.0116 USD per Hour

On-Demand Windows pricing: 0.0162 USD per Hour

Free tier eligible

Compare instance types

4) Select Instance Type

## ▼ Key pair (login) [Info](#)

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

Key pair name

Don't include in launch template

 [Create new key pair](#)

## ▼ Network settings

Networking platform [Info](#)

**Virtual Private Cloud (VPC)**

Launch into a virtual network in your own logically isolated area within the AWS Cloud

**EC2-Classic**

Launch into a single flat network that you share with other customers.

Security groups

Select security groups

my-lab-EC2-Server-sg sg-0a370c15c5b405b61   
VPC: vpc-0b978358e22761686



5) Select Server security Group you Created

## Create Auto Scaling Group

### Go to Auto Scaling Display from EC2 Display

Snapshots  
Lifecycle Manager  
▼ Network & Security  
Security Groups  
Elastic IPs  
Placement Groups  
Key Pairs  
Network Interfaces  
▼ Load Balancing  
Load Balancers  
Target Groups New  
▼ Auto Scaling  
Launch Configurations  
Auto Scaling Groups

**Resources**

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

Instances (running)	0	Dedicated Hosts	0
Elastic IPs	0	Instances	0
Key pairs	4	Load balancers	1
Placement groups	0	Security groups	9
Snapshots	0	Volumes	0

Facilitate configuration and deployment Microsoft SQL Server Always On availability groups for SQL Server. [Learn more](#)

### Create Auto Scaling Group

**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 failover of your application.

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

**Create Auto Scaling group**

...

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** 2) Name 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** 3) Select Your Launch Template

Create a launch template

Version  
Default (1) C

**Create a launch template version Info**

Description -	Launch template <b>my-lab-server</b> <span style="border: 1px solid red; padding: 2px;">4) Click Next</span> lt-0ca69b78349a3fb83	Instance type t2.micro
AMI ID ami-0dc2d3e4c0f9ebd18	Security groups -	Request Spot Instances No
Key pair name -	Security group IDs <b>sg-0a370c15c5b405b61</b>	
Additional details		
Storage (volumes) -	Date created Sun Jul 11 2021 11:24:24 GMT-0500 (Central Daylight Time)	<span style="border: 1px solid red; padding: 2px;">Cancel</span> <span style="background-color: orange; color: white; border: 1px solid orange; padding: 2px;">Next</span>

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

## 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 C 5) Select VPC

Create a VPC

Subnets  
Select subnets C

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

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

## 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

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.

**8) Select Attach to Existing Load Balancer**

**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**

**10) Click Next**

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

Cancel Previous Skip to review **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 group size and scaling policies Info

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

### Group size - optional Info

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

Desired capacity

2

Minimum capacity

1

Maximum capacity

3

**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

Target Tracking Policy

Metric type

Average CPU utilization

Target value

50

**12) Set Target Tracking for CPU Utilization**

Instances need

300 seconds warm up before including in metric

Disable scale in to create only a scale-out policy

### Instance scale-in protection - optional

Instance scale-in protection

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

Enable instance scale-in protection

**13) Click Next**

Cancel

Previous

Skip to review

Next

...

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

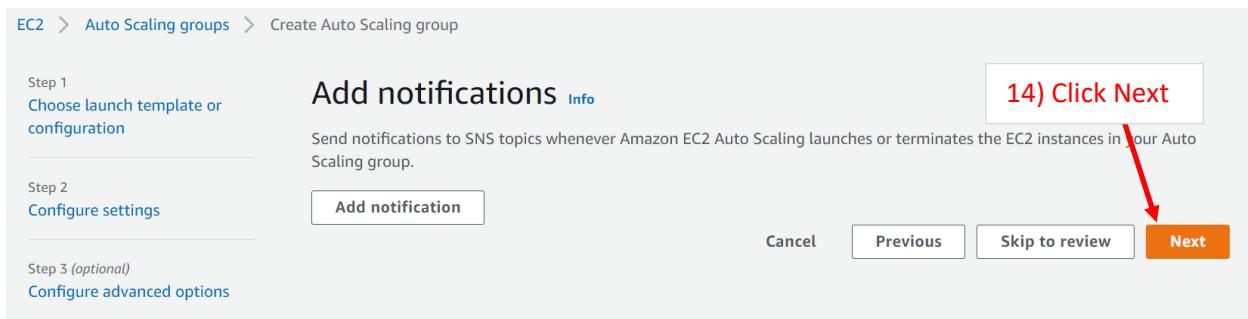
Add notifications Info

Send notifications to SNS topics whenever Amazon EC2 Auto Scaling launches or terminates the EC2 instances in your Auto Scaling group.

Add notification

Cancel Previous Skip to review Next

**14) Click Next**



...

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

Add tags Info

Add tags to help you search, filter, and track your Auto Scaling group across AWS. You can also choose to automatically add these tags to instances when they are launched.

ⓘ You can optionally choose to add tags to instances (and their attached EBS volumes) by specifying tags in your launch template. We recommend caution, however, because the tag values for instances from your launch template will be overridden if there are any duplicate keys specified for the Auto Scaling group.

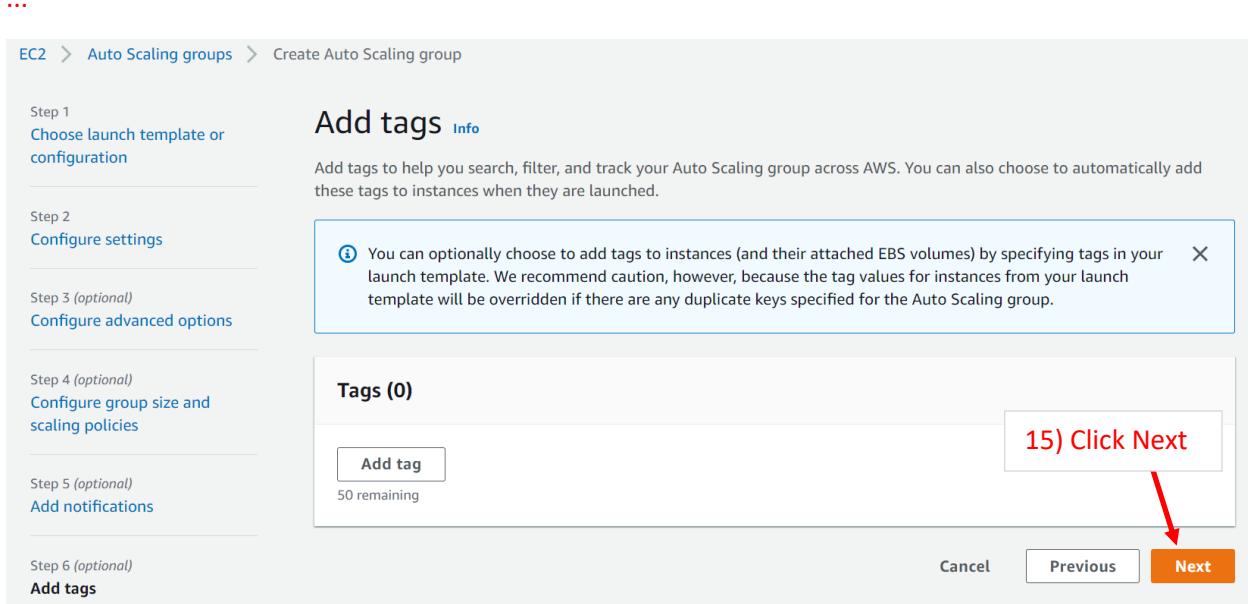
Tags (0)

Add tag

50 remaining

Cancel Previous Next

**15) Click Next**



...

EC2 > Auto Scaling groups > Create Auto Scaling group

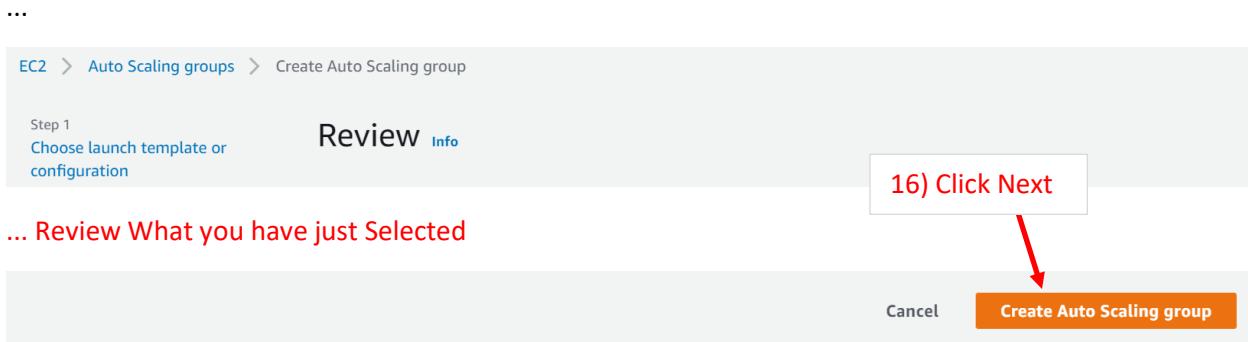
Step 1 Choose launch template or configuration

Review Info

... Review What you have just Selected

Cancel Create Auto Scaling group

**16) Click Next**



## Verify and Test the ALB

View the Health Check on your the Target Group Details. Both Instances Should be Healthy

my-lab-target Delete

arn:aws:elasticloadbalancing:us-east-1:409673912482:targetgroup/my-lab-target/785ca90756d47acd

Details					
Target type Instance	Protocol : Port HTTP: 80	Protocol version HTTP1	VPC <a href="#">vpc-0b978358e22761686</a>		
Load balancer <a href="#">my-lab-alb</a>					
Total targets 2	Healthy <a href="#">2</a>	Unhealthy <a href="#">0</a>	Unused <a href="#">0</a>	Initial <a href="#">0</a>	Draining <a href="#">0</a>

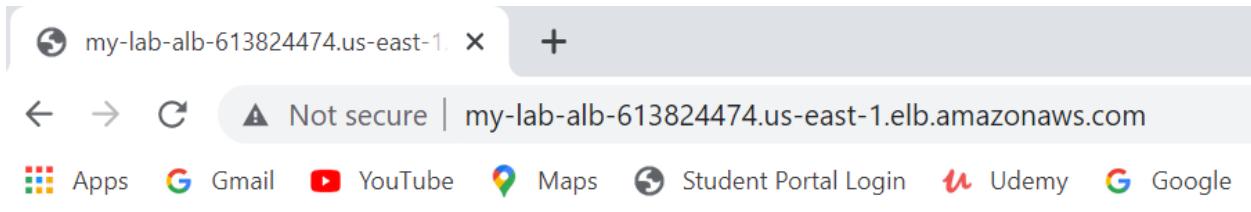
**Targets** Monitoring Health checks Attributes Tags

Registered targets (2)					<a href="#">C</a>	Deregister	Register targets
<input type="text"/> Filter resources by property or value					<a href="#">&lt;</a>	<a href="#">1</a>	<a href="#">&gt;</a>
<input type="checkbox"/>	Instance ID	Name	Port	Zone	Health status	Health status details	
<input type="checkbox"/>	<a href="#">i-0179bc9cdca16967e</a>		80	us-east-1b	<a href="#">健康的</a>	<a href="#">healthy</a>	
<input type="checkbox"/>	<a href="#">i-0f05d00a3423df3ad</a>		80	us-east-1a	<a href="#">健康的</a>	<a href="#">healthy</a>	

DNS on Load Balancer Display. Each EC2 will have a public address but you cannot access due to security group settings.

The screenshot shows the AWS CloudFormation console interface. At the top, there are buttons for 'Create Load Balancer' and 'Actions ▾'. Below that is a search bar with the placeholder 'Filter by tags and attributes or search by keyword'. The main area has a table with the following columns: Name, DNS name, State, and VPC ID. One row is visible, showing 'my-lab-alb' as the Name, 'my-lab-alb-613824474.us-east-1.elb.amazonaws.com' as the DNS name, 'Active' as the State, and 'vpc-0b978358e22761686' as the VPC ID. A red arrow points to the 'DNS' tab in the top navigation bar.

Test DNS with Web Browser



## EC2 stress tool

1-select the EC2 instance you want to install the stress tool: we can use the instance we have during the ASG class.

install stress tool using the following commands:

```
sudo amazon-linux-extras install epel -y
```

```
sudo yum install stress -y
```

```
Dependencies Resolved

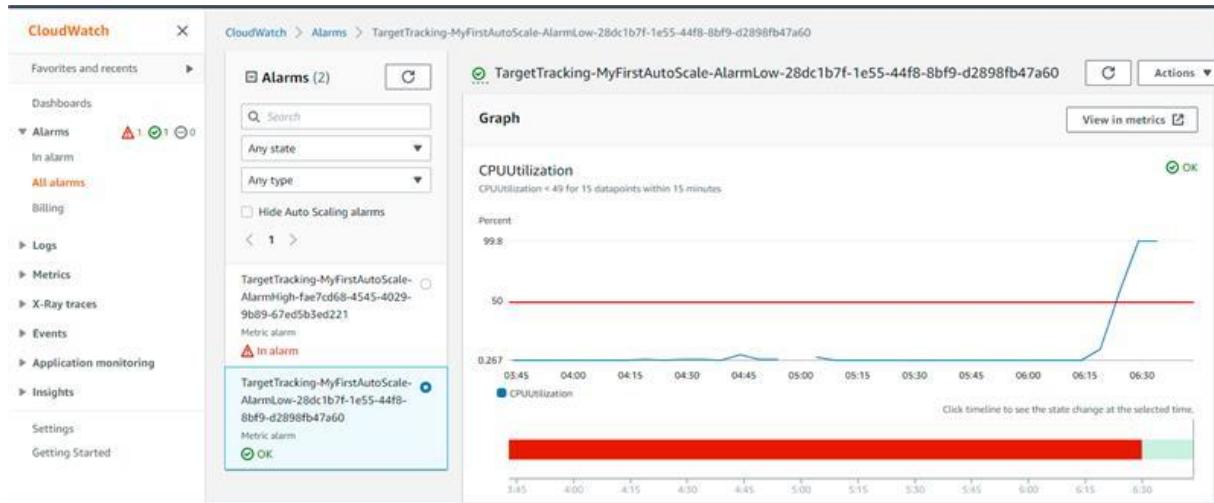
Package           Arch      Version       Repository      Size
stress            x86_64   1.0.4-16.el7    epel           39 k

Transaction Summary
install 1 Package

total download size: 39 k
downloaded size: 34 k
downloading packages:
stress-1.0.4-16.el7.x86_64.rpm: Header V3 RSA/SHA256 Signature, key ID 352e64e5; NOKEY
Public key for stress-1.0.4-16.el7.x86_64.rpm is not installed
stress-1.0.4-16.el7.x86_64.rpm
Retrieving GPG key from /etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-7
Importing GPG key 0x152c64e5:
Userid : "Fedora EPEL (7) <epel@fedoraproject.org>"
Fingerprint: 91e9 7d7c 4a5e 9ef1 7d1e 000a 6a2f ae2 352c 64e5
Packag
  : epel-release-7-11.noarch (8mmn2extra-epel)
From   : /etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-7
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : stress-1.0.4-16.el7.x86_64
  Verifying  : stress-1.0.4-16.el7.x86_64
Installed:
  stress.x86_64 0:1.0.4-16.el7
complete!
```

Then to visualize the CPU and memory utilization write the following commands:

```
sudo stress --cpu 8 --vm-bytes $(awk '/MemAvailable/{printf "%d\n", $2 * 0.9;}' < /proc/meminfo)k --vm-keep -m 1
```



-cpu

This will spawn 8 CPU workers spinning on a square root task ( $\sqrt{x}$ )

-vm-bytes

This will use 90% of the available memory from /proc/meminfo

-vm-keep

This will re-dirty memory instead of freeing and reallocating.

-m 1

This will spawn 1 worker spinning on malloc()/free()

As time goes on, it will continue to update the graph. To remove the load, press

CTRL-C to stop the stress script.

Reference: [https://www.wellarchitectedlabs.com/performance-efficiency/100\\_labs/100\\_monitoring\\_linux\\_ec2](https://www.wellarchitectedlabs.com/performance-efficiency/100_labs/100_monitoring_linux_ec2)