

Assignment 4 – ALB and ASG

ACCEPTANCE CRITERIA – Include the followings in the PDF:

- 2 links for ALB and NLB.
- Screenshot of healthy instances in TG for.
- 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. Practice Listener rules with 2 lambdas

- a. Create 2 Lambdas in the default **VPC**. The first server returns “App 1” and the second server returns “App2”.
- b. Create a “TG1” and register the “App 1” lambda. Create a “TG2” and register the “App2” lambda.
- c. Create the ALB.
 - i. The default rule can be any TG.
 - ii. If the request path starts with “app1”, the route the to TG1 (App1)
 - iii. If the request path starts with “app2”, the route the to TG2 (App2)

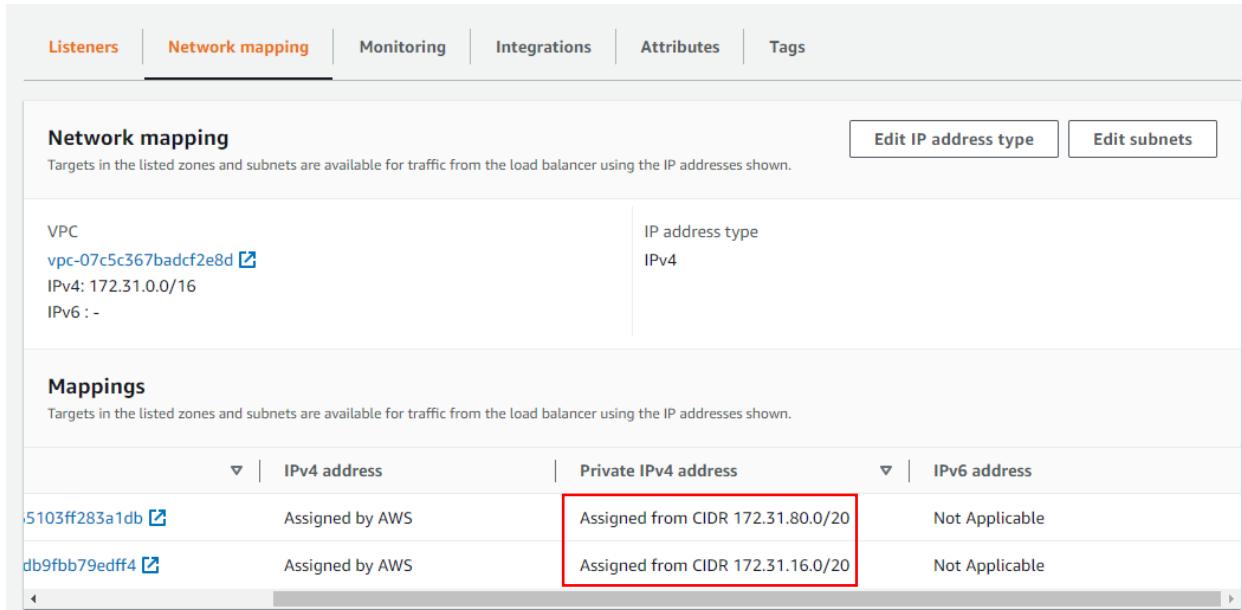
Task 3. Run web servers behind NLB

NLB takes 5 minutes to be provisioned. So please wait patiently.

NLB operates at layer 4 and has fewer features even though it is hyper-performant. You can not associate NLB with SG. How you are going to whitelist access only from NLB in your EC2?

The solution is to whitelist subnets (172.31.x.x) in which the NLB nodes are created.

- a. Add the instances you created in task 1 to the target group of the NLB. The protocol must be **TCP (Layer 4)**, not HTTP (Layer 4).
- b. Once NLB is provisioned, you will find the subnets in which the NLB nodes are created. Whitelist them in the web server SG.



IPv4 address	Private IPv4 address	IPv6 address
Assigned by AWS	Assigned from CIDR 172.31.80.0/20	Not Applicable
Assigned by AWS	Assigned from CIDR 172.31.16.0/20	Not Applicable

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

By default, your servers see the clients' IP addresses. We don't want that. Because we want to allow access only from the NLB in the web servers. For that, you must deselect "Preserve client IP addresses". So, your servers see the NLB nodes' IP address as the source IP instead of the clients' IPs when making a call with the URL.

Task 4. Run the web server behind the ALB 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 5. 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 <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
Owner <input type="checkbox"/> 409673912482	Inbound rules count 1 Permission entry	Outbound rules count 1 Permission entry	

Inbound rules Outbound rules Tags

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 <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
Owner <input type="checkbox"/> 409673912482	Inbound rules count 1 Permission entry	Outbound rules count 1 Permission entry	

Inbound rules Outbound rules Tags

Inbound rules (1/1)

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 the following sections:

- 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
- ...

The main content 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
Volumes	0

A red arrow points from the "Load Balancers" link in the sidebar to the "Load balancers" count in the Resources table. A red box highlights the "Load balancers" row with the text "1) Click on Load Balancers".

The screenshot shows the "Load Balancers" display page. At the top, there is a header with a "Create Load Balancer" button (highlighted with a red arrow), an "Actions" dropdown, and other navigation icons.

The main content area has a search bar with the placeholder "Filter by tags and attributes or search by keyword". Below the search bar, there are filter options for Name, DNS name, State, VPC ID, and Availability Zones.

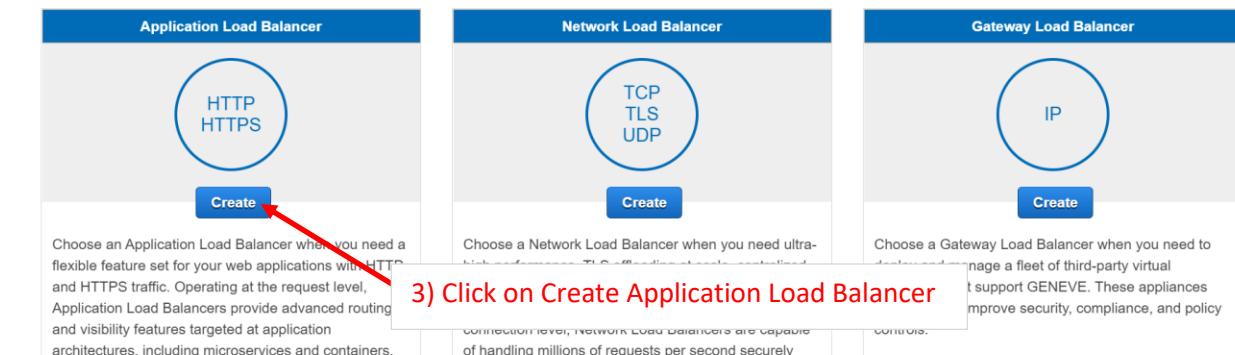
A red box highlights the "Create Load Balancer" button with the text "2) Click on Create Load Balancer".

The message "You do not have any load balancers in this region." is displayed at the bottom of the page.

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](#)



1. Configure Load Balancer 2. Configure Security Settings 3. Configure Security Groups 4. Configure Routing 5. Register Targets 6. Review

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"/>
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	vpc-0b978358e22761686 my-lab-vpc
Availability Zones	<input checked="" type="checkbox"/> us-east-1a <input type="text" value="subnet-0ef43ef1cfcb561a0 (lab-sn-public-1A)"/> <input checked="" type="checkbox"/> us-east-1b <input type="text" value="subnet-03b7f8298553c4646 (lab-sn-public-1B)"/>

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

...
1. Configure Load Balancer 2. Configure Security Settings 3. Configure Security Groups 4. Configure Routing 5. Register Targets 6. Review

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

...
1. Configure Load Balancer 2. Configure Security Settings 3. Configure Security Groups 4. Configure Routing 5. Register Targets 6. Review

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

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

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"/>	10) Name 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	11) Select Instance
Protocol	<input type="text" value="HTTP"/>	
Port	<input type="text" value="80"/>	
Protocol version	<input checked="" type="radio"/> HTTP1 <small>Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.</small>	

Health checks

Protocol	<input type="text" value="HTTP"/>
Path	<input type="text" value="/"/>

► 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.

<input type="button" value="Remove"/>	<input type="checkbox"/>	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.

[Add to registered](#) on port

13) Click Next

[Cancel](#) [Previous](#) **Next: Review**

1. Configure Load Balancer 2. Configure Security Settings 3. Configure Security Groups 4. Configure Routing 5. Register Targets **6. 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-0ef43eff1cfcb561a0 (lab-sn-public-1A), subnet-03b7f8298553c4646 (lab-sn-public-1B)
Tags

▼ Security groups

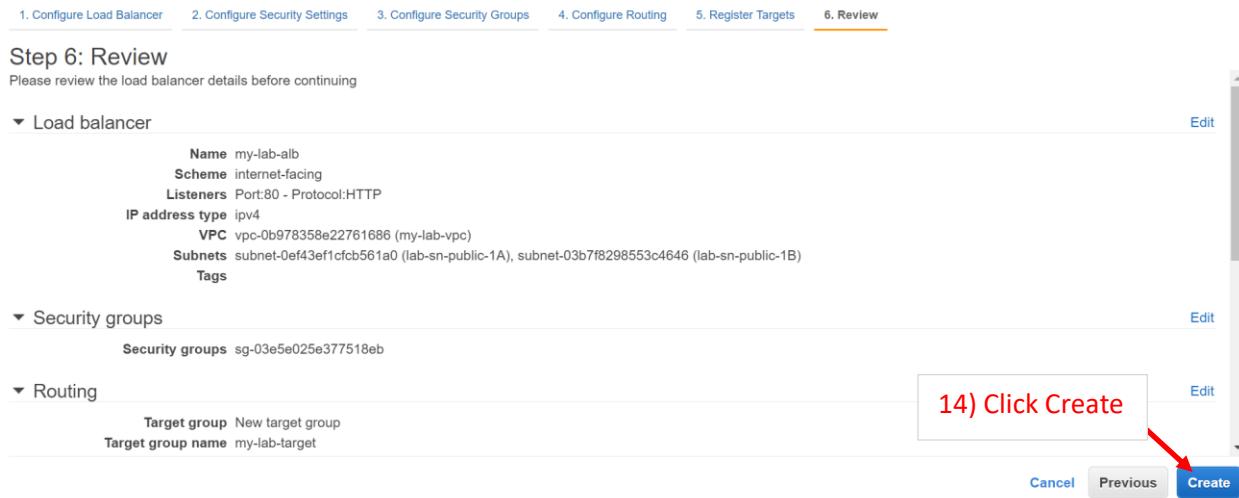
Security groups sg-03e5e025e377518eb

▼ Routing

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

14) Click Create

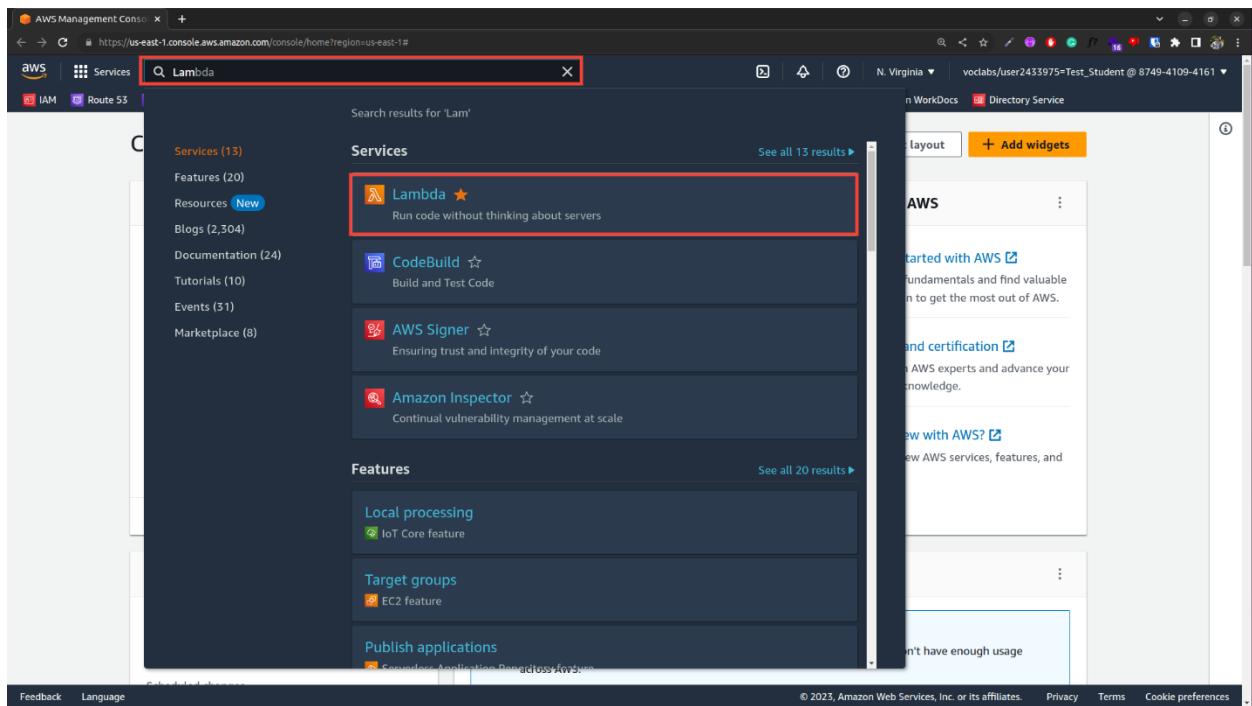
Cancel Previous **Create**



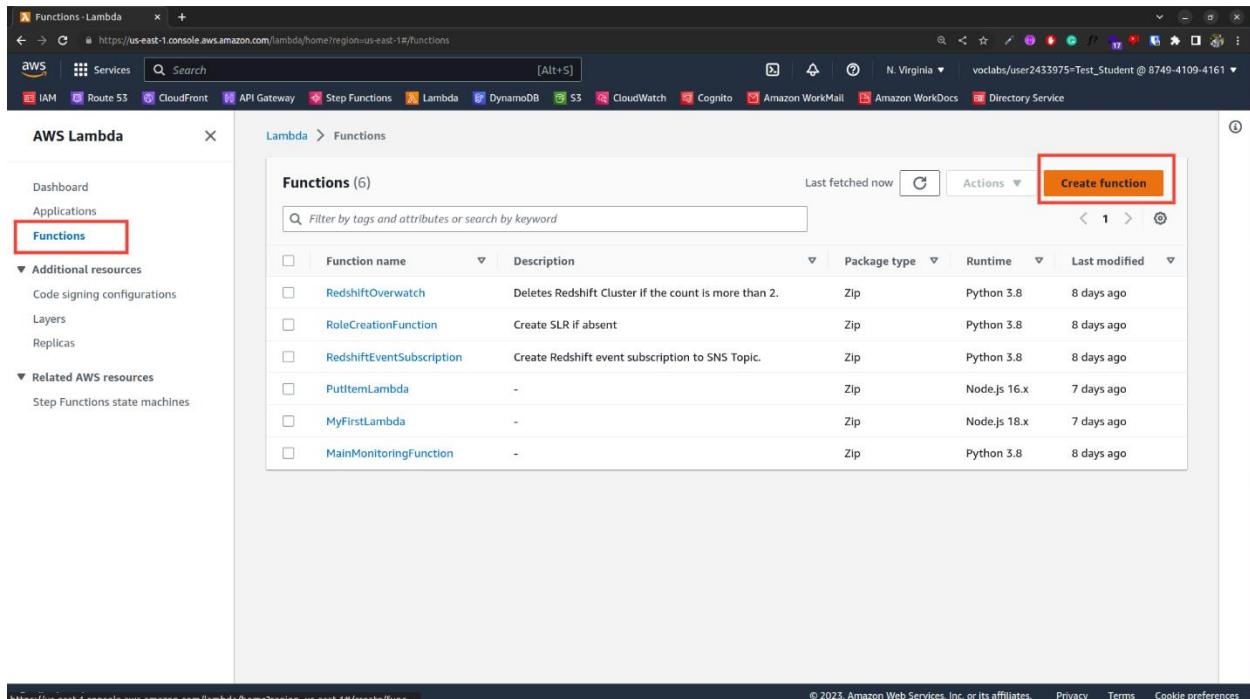
Task 2. Practice Listener rules with 2 lambdas

Part-1: Creating Lambda Function

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



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



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 [Info](#)

Cancel **Create function**

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**

Advanced settings

Enable Code signing [Info](#)
Use code signing configurations to ensure that the code has been signed by an approved source and has not been altered since signing.

Enable function URL [Info](#)
Use function URLs to assign HTTPS endpoints to your Lambda function.

Auth type
 AWS_IAM Only authenticated IAM users and roles can make requests to your function URL.
 NONE Lambda won't perform IAM authentication on requests to your function URL. The URL endpoint will be public unless you implement your own authorization logic in your function.

Function URL permissions

When you choose auth type NONE, Lambda automatically creates the following resource-based policy and attaches it to your function. This policy makes your function public to anyone with the function URL. You can edit the policy later. To limit access to authenticated IAM users and roles, choose auth type AWS_IAM.

[View policy statement](#)

Configure cross-origin resource sharing (CORS)
Use CORS to allow access to your function URL from any origin. You can also use CORS to control access for specific HTTP headers and methods in requests to your function URL. By default, all origins are allowed. You can edit this after creating the function. [Learn more](#)

Enable tags [Info](#)
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources, track your AWS costs, and enforce attribute-based access control.

Enable VPC [Info](#)
Connect your function to a VPC to access private resources during invocation.

[Cancel](#) **Create function**

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;
};
```

```

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,
10     body: JSON.stringify('Hello from app1')
11   };
12   return response;
13 }
14
15

```

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.

Name	Security group ID	Security group name	VPC ID	Description	Owner	Inbound rules count
-	sg-050445e73404dfa0d	default	vpc-0f5cdad7232fa1fda	default VPC security gr...	874941094161	1 Permission entry
-	sg-03e4fe449abe6a665	ForWebServer	vpc-0f5cdad7232fa1fda	ForWebServer	874941094161	2 Permission entries
-	sg-aa0a479396f622d10	ForAlb	vpc-0f5cdad7232fa1fda	ForAlb	874941094161	1 Permission entry
-	sg-0acaad4d5e6515bf97	launch-wizard-1	vpc-0f5cdad7232fa1fda	launch-wizard-1 create...	874941094161	1 Permission entry
-	sg-06e0254b6ed2495eb	launch-wizard-2	vpc-0f5cdad7232fa1fda	launch-wizard-2 create...	874941094161	1 Permission entry
-	sg-02e8c0193fd46c318	load-balancer-sg	vpc-0f5cdad7232fa1fda	load-balancer-sg	874941094161	1 Permission entry
-	sg-098d5cea946e63944	default	vpc-06067a93b2b537752 ...	default VPC security gr...	874941094161	1 Permission entry
-	sg-0a829b05fe5a8ac86	Alb-Lambda-sg	vpc-0f5cdad7232fa1fda	Alb-Lambda-sg	874941094161	1 Permission entry
-	sg-0ef184db695599715	web-servers-sg	vpc-0f5cdad7232fa1fda	for web servers runnin...	874941094161	3 Permission entries
-	sg-02e86d22c02598fae	launch-wizard-3	vpc-0f5cdad7232fa1fda	launch-wizard-3 create...	874941094161	0 Permission entries

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' page in the AWS Management Console. Under 'Basic details', the 'Security group name' is set to 'Alt-Lambda-sg' and the 'Description' is also 'Alt-Lambda-sg'. A VPC is selected with the ID 'vpc-0f5cdad7232fa1fda'. In the 'Inbound rules' section, a new rule is being configured with 'Type' set to 'HTTP', 'Protocol' to 'TCP', and 'Port range' to '80'. The 'Source' dropdown is set to 'Anywhere...'. The 'CIDR block' field contains '0.0.0.0/0'. An 'Add rule' button is visible below the table. The 'Outbound rules' section is currently empty.

2. c Click on Create Security Group.

The screenshot shows the 'Create security group' page with the following configurations: 'Type' is 'HTTP', 'Protocol' is 'TCP', 'Port range' is '80', 'Source' is 'Anywhere...', and 'CIDR block' is '0.0.0.0/0'. Below this, the 'Outbound rules' section shows a rule for 'All traffic' with 'Protocol' 'All', 'Port range' 'All', and 'Destination' 'Custom', with 'CIDR block' '0.0.0.0/0'. The 'Tags - optional' section is empty. At the bottom right, there is a 'Cancel' button and a prominent orange 'Create security group' button, which is highlighted with a red box.

Part-3: Create Target Group TG1 and TG2.

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

The screenshot shows the AWS EC2 Target Groups page. On the left, there's a navigation sidebar with various services like 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, 'Target Groups' is selected and highlighted with a red box. The main content area shows a table titled 'Target groups (6) Info'. A new target group, 'TG1', has just been created and is listed at the bottom of the table. The 'Create target group' button is also highlighted with a red box. The table columns include Name, ARN, Port, Protocol, Target type, Load balancer, and VPC ID.

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

The screenshot shows the 'Specify group details' step of the Lambda target group creation wizard. It's divided into two sections: 'Step 1 Specify group details' and 'Step 2 Register targets'. In Step 1, under 'Basic configuration', it says 'Your load balancer routes requests to the targets in a target group and performs health checks on the targets.' Below this, the 'Choose a target type' section is shown. The 'Lambda function' option is selected and highlighted with a red box. The 'Target group name' field contains 'TG1', which is also highlighted with a red box. The 'Basic configuration' section notes that settings cannot be changed after the target group is created.

3.c Click Next.

Target group name
TG1

Health checks
The associated load balancer periodically sends requests, per the settings below, to the registered targets to test their status.

Enable
Health checks count as a request for your Lambda function. Refer to Lambda pricing for more details [details](#)

Attributes

Certain default attributes will be applied to your target group. You can view and edit them after creating the target group.

Tags - optional
Consider adding tags to your target group. Tags enable you to categorize your AWS resources so you can more easily manage them.

Cancel **Next Step**

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

Step 1
Specify group details

Step 2
Register targets

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.

Lambda function
You can specify a single Lambda function as the target

Choose a Lambda function from the list or [create function](#) [Details](#)
App1

Version Alias
\$LATEST

Enter a Lambda function ARN. [Lambda](#) [Details](#)
 Add a function later

Cancel Previous **Create target group**

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 AWS Management Console with the EC2 service selected. In the left sidebar, under the 'Load Balancing' section, the 'Load Balancers' option is highlighted. The main pane displays a table of three existing load balancers: 'my-nlb', 'my-alb', and 'AlbDemo'. Each row includes columns for Name, DNS name, State, VPC ID, Availability Zones, Type, Date created, and Instance. A red box highlights the 'Create load balancer' button at the top right of the table.

4. b Select Create from Application Load Balancer.

The screenshot shows the 'Create Load Balancer' dialog box. It compares three types of load balancers: Application Load Balancer, Network Load Balancer, and Gateway Load Balancer. The 'Application Load Balancer' section is selected, showing a diagram where traffic enters through an ALB, which then routes it to three targets (Lambda, API Gateway, and Container). It supports HTTP and HTTPS. A red box highlights the 'Create' button. The other sections show similar diagrams and descriptions for Network and Gateway load balancers.

4. c Fill out the Load balancer name.

Create Application Load Balancer Info

The Application Load Balancer distributes incoming HTTP and HTTPS traffic across multiple targets such as Amazon EC2 Instances, microservices, and containers, based on request attributes. When the load balancer receives a connection request, it evaluates the listener rules in priority order to determine which rule to apply, and if applicable, it selects a target from the target group for the rule action.

▶ How Elastic Load balancing works

Basic configuration

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

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-0f0ccad7232fa1da
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-05e3088a6bc6a72d2
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.

The screenshot shows the AWS CloudFront console. In the top navigation bar, 'Services' is selected. Below it, 'CloudFront' is highlighted. A search bar and a 'Create new security group' button are visible. A dropdown menu titled 'Select up to 5 security groups' contains one item: 'Alb-Lambda-sg sg-0a829b05fe3a8ac86'. This item is highlighted with a red box. The 'Listeners and routing' section is expanded, showing a table for Listener HTTP:80. The first row has 'Protocol: HTTP', 'Port: 80', and 'Default action: Info'. The 'Forward to' dropdown is set to 'Select a target group' and is also highlighted with a red box. Below this, there's a 'Listener tags - optional' section with a 'Add listener tag' button.

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

This screenshot is identical to the previous one, but the 'Forward to' dropdown in the Listener table now contains 'TG1 Target type: Lambda, IPv4'. This selection is highlighted with a red box. The rest of the interface, including the security group list and the 'Listeners and routing' section, remains the same.

4. g Click on Create load balancer.

The screenshot shows the 'Create Load Balancer' wizard. The 'Basic configuration' section includes 'ALB-Lambda' under 'Name', 'Internet-facing' under 'Type', and 'IPv4' under 'IP Version'. The 'Security groups' section lists 'Alb-Lambda-sg' and its VPC ID 'vpc-0f5cdad7232fa1fda'. The 'Network mapping' section shows 'HTTP:80' mapped to 'TG1'. The 'Listeners and routing' section shows 'HTTP:80' defaults to 'TG1'. The 'Tags' section is empty. The 'Attributes' section contains a note about default attributes. At the bottom right is a 'Create load balancer' button.

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

The screenshot shows the 'Load balancers' page under the EC2 service. It lists four load balancers: 'my-nlb', 'my-alb', 'AlbDemo', and 'ALB-Lambda'. The 'ALB-Lambda' row is highlighted with a red box. The table columns include Name, DNS name, State, VPC ID, Availability Zones, Type, Date created, and Instance. A 'Create load balancer' button is visible at the top right of the table area.

	Name	DNS name	State	VPC ID	Availability Zones	Type	Date created	Instance
<input type="checkbox"/>	my-nlb	my-nlb-f327d7fce235b24...	Active	vpc-0f5cdad7232fa1fda	2 Availability Zones	network	March 2, 2023, 13:44 (UTC-06:00)	-
<input type="checkbox"/>	my-alb	my-alb-355457061.us-eas...	Active	vpc-0f5cdad7232fa1fda	2 Availability Zones	application	March 2, 2023, 10:58 (UTC-06:00)	-
<input type="checkbox"/>	AlbDemo	AlbDemo-960472562.us-e...	Active	vpc-0f5cdad7232fa1fda	2 Availability Zones	application	March 3, 2023, 14:03 (UTC-06:00)	-
<input type="checkbox"/>	ALB-Lambda	ALB-Lambda-1678675122...	Active	vpc-0f5cdad7232fa1fda	2 Availability Zones	application	March 6, 2023, 21:26 (UTC-06:00)	-

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

Screenshot of the AWS CloudFront console showing the configuration for an Application Load Balancer (ALB). The 'Listeners' tab is selected. A rule for port 80 is listed, forwarding traffic to Target Group TG1. The 'Manage rules' button in the Actions menu is highlighted with a red box.

4. j Click on Edit Icon.

Screenshot of the AWS CloudFront rule editor for the ALB-Lambda | HTTP:80 listener. The 'Edit' icon in the top left corner is highlighted with a red box. The rule configuration shows an IF condition 'Requests otherwise not routed' and a THEN action 'Forward to TG1: 1 (100%)'. The 'Last' action is also visible.

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

The screenshot shows the AWS Lambda function configuration page for an ALB-Lambda target. In the 'Edit Rule' section, there is a dropdown menu under the 'THEN' column. The option 'Return fixed response...' is highlighted with a red box. The 'Update' button is located at the top right of the rule editor.

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 for an ALB-Lambda target. The 'Edit Rule' section has been updated. The 'Response code' field contains '200', and the 'Response body' field contains 'Fixed Response'. The 'Update' button 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 Rules configuration interface. At the top, there's a navigation bar with various services like IAM, Route 53, CloudFront, API Gateway, Step Functions, Lambda, DynamoDB, S3, CloudWatch, Cognito, Amazon WorkMail, Amazon WorkDocs, and Directory Service. Below the navigation bar, the main title is 'ALB-Lambda | HTTP:80'. On the left, there's a sidebar with a 'Rules' section. In the center, there's a table for defining rules. The first rule is named 'HTTP 80: default action' and has an 'IF' condition 'Requests otherwise not routed'. The 'THEN' action is 'Return fixed response 200 (more...)'. At the top right of the rule table, there's a red box around the 'Insert Rule' button.

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

This screenshot shows the same AWS Lambda Rules configuration interface as the previous one, but with a different focus. A red box highlights the 'Path...' option in the 'Add condition' dropdown menu. The rule table shows one rule named 'HTTP 80: default action' with the 'Path...' condition selected. The 'THEN' action is 'Return fixed response 200 (more...)'. The 'Save' button is visible at the top right.

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

The screenshot shows the AWS Lambda function configuration page for 'ALB-Lambda | HTTP:80'. A single rule is defined:

- RULE ID:** 1
- IF (all match):** Path... is `/app1`
- THEN:** Forward to... (selected)

Note: Additional actions are available for HTTPS listeners.

Below the rule table, it says: 'last' **HTTP 80: default action** **IF** Requests otherwise not routed. This rule cannot be moved or deleted.

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

The screenshot shows the AWS Lambda function configuration page for 'ALB-Lambda | HTTP:80'. The rule has been modified:

- RULE ID:** 1
- IF (all match):** Path... is `/app1`
- THEN:** 1. Forward to... (selected)
 - Target group : Weight (0-999) **TG1** (selected)
 - Traffic distribution 100%
 - Select a target group 0
 - Group-level stickiness (checked)

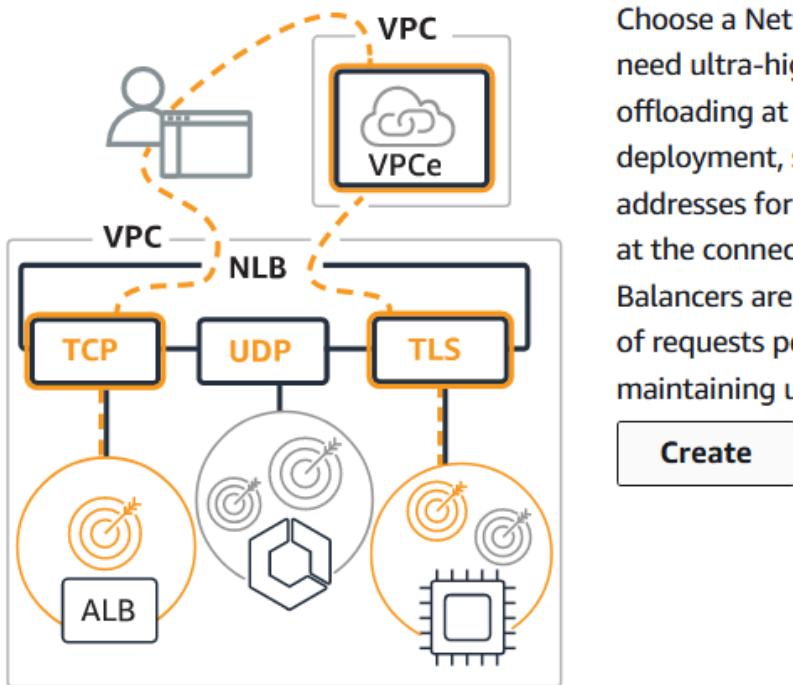
Note: Additional actions are available for HTTPS listeners.

Below the rule table, it says: 'last' **HTTP 80: default action** **IF** Requests otherwise not routed. This rule cannot be moved or deleted.

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

Task 3. Run web servers behind 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
-	i-0452e560715ee832a	Running Q Q	t2.micro	2/2 checks passed	No alarms	+ us-east-1a
<input checked="" type="checkbox"/> myNlbE1	i-03f9925915e9136a6	Running Q Q	t2.micro	Initializing	No alarms	+ us-east-1a
<input checked="" type="checkbox"/> myNLBE2	i-009a5fdf725c0d8d2	Running Q Q	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

mySgNlb

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

Protocol

Port

TCP : 80

VPC

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

my-first-vpc
vpc-0def861cf2ef04f24
IPv4: 10.0.0.0/16

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		running	MyAppBehindAlb	us-east-1a	subnet-0b5aeb0697c77d6a5
<input checked="" type="checkbox"/>	i-03f9925915e9136a6	myNLbE1	running	default	us-east-1a	subnet-0b5aeb0697c77d6a5
<input checked="" type="checkbox"/>	i-009a5fdf725c0d8d2	myNLbE2	running	default	us-east-1b	subnet-0d2438927d9c121

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

Edit attributes

Attributes	Restore defaults
<p>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. <input type="text" value="300"/> seconds 0-3600</p> <p><input type="checkbox"/> Connection termination on deregistration — recommended If enabled, your Network Load Balancer will terminate active connections when deregistration delay is reached.</p> <p><input type="checkbox"/> 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.</p> <p><input type="checkbox"/> 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.</p> <p><input type="checkbox"/> Preserve client IP addresses Preserve client IP addresses and ports in the packets forwarded to targets.</p>	

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.

VPC vpc-07c5c367badcf2e8d [edit] 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
5103ff283a1db [edit]	Assigned by AWS	Assigned from CIDR 172.31.80.0/20	Not Applicable
db9fbb79edff4 [edit]	Assigned by AWS	Assigned from CIDR 172.31.16.0/20	Not Applicable

Inbound rules (4)

[Filter security group rules](#)

Rule ID	IP version	Type	Protocol	Port range	Source	Description
0dca9daad	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	-

Task 4. Run the web server behind the ALB in ASG

Go to Launch Templates Display from EC2 Display

New EC2 Experience X

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

1) Click Create Launch Template

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

...

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

The screenshot shows the AWS EC2 Resources page. On the left, there is a navigation sidebar with the following items:

- Snapshots
- Lifecycle Manager
- Network & Security** (expanded)
 - Security Groups
 - Elastic IPs
 - Placement Groups
 - Key Pairs
 - Network Interfaces
- Load Balancing** (expanded)
 - Load Balancers
 - Target Groups New
- Auto Scaling** (expanded)
 - Launch Configurations
 - Auto Scaling Groups**

The main content area is titled "Resources" and displays the following resource summary:

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

A callout box with the text "1) Click on Auto Scaling Groups" points to the "Auto Scaling Groups" link in the sidebar.

Create Auto Scaling Group

The screenshot shows the "Create Auto Scaling group" landing page. The main heading is "Amazon EC2 Auto Scaling helps maintain the availability of your applications". Below the heading, there is a subtext: "Auto Scaling groups are collections of Amazon EC2 instances that enable automatic scaling and free up capacity for your application." A callout box with the text "1) Click on Create Auto Scaling Group" points to the "Create Auto Scaling group" button.

...

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 my-lab-server 4) Click Next	Instance type t2.micro
AMI ID 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

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

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

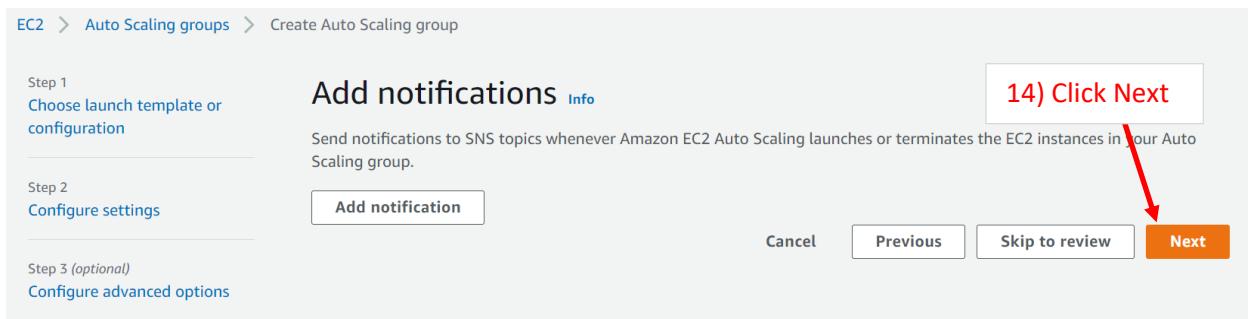
Add notifications Info

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.

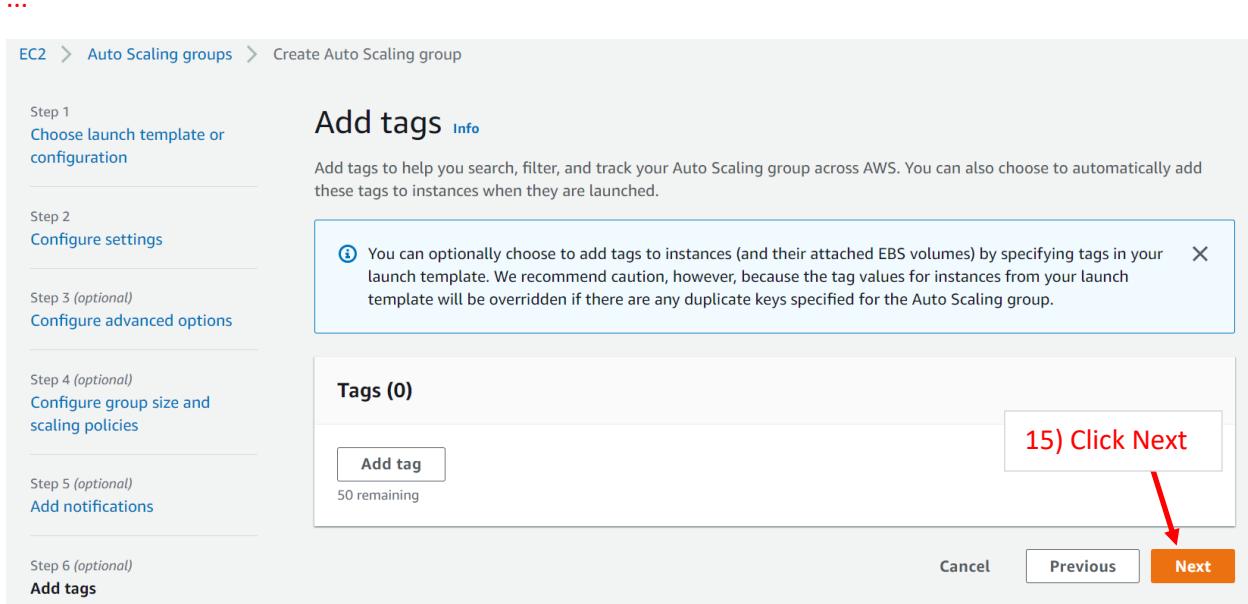
Tags (0)

Add tag

50 remaining

Cancel Previous Next

15) Click Next



...

EC2 > Auto Scaling groups > Create Auto Scaling group

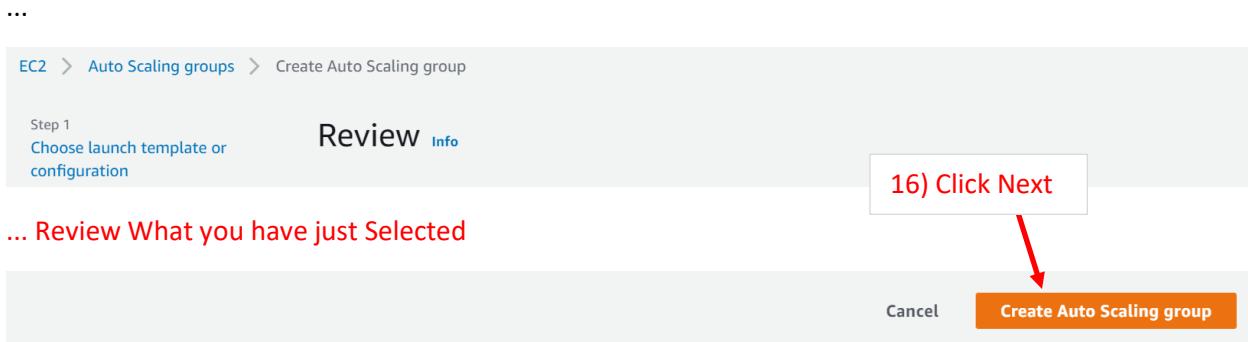
Step 1 Choose launch template or configuration

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 vpc-0b978358e22761686		
Load balancer my-lab-alb					
Total targets 2	Healthy ✓ 2	Unhealthy ✗ 0	Unused ○ 0	Initial ⌚ 0	Draining ⊖ 0

Targets Monitoring Health checks Attributes Tags

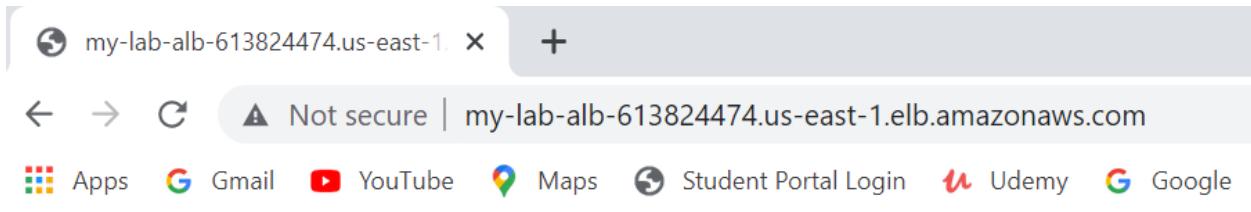
Registered targets (2)					C	Deregister	Register targets
<input type="text"/> Filter resources by property or value					<	1	>
	Instance ID	Name	Port	Zone	Health status	Health status details	
<input type="checkbox"/>	i-0179bc9cdca16967e		80	us-east-1b	✓ healthy		
<input type="checkbox"/>	i-0f05d00a3423df3ad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

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



DNS	
Name	DNS name
my-lab-alb	my-lab-alb-613824474.us-east-1.elb.amazonaws.com

Test DNS with Web Browser



Hello from my EC2 Instance in Autoscaling Group Behind an ALB

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

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
```



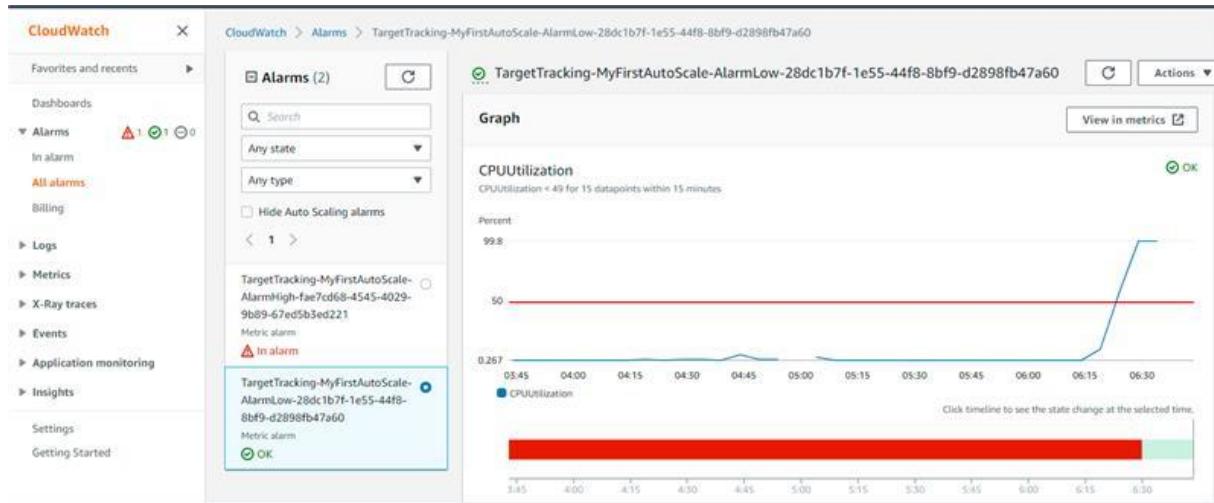
```
Dependencies Resolved

Package           Arch      Version       Repository      Size
Installing:
stress            x86_64    1.0.4-16.el7   epel           39 k
Transaction Summary
Install 1 Package

Total download size: 39 k
Installed size: 94 k
Downloading packages:
warning: /var/cache/yum/x86_64/2/epel/packages/stress-1.0.4-16.el7.x86_64.rpm: Header V3 RSA/SHA256 Signature, key ID 352c64e5: 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 key from file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-7
Importing GPG key 0x352c64e5:
Userid : "Fedora EPEL (7) <epel@fedoraproject.org>"
Fingerprint: 91e9 7d7c 4a5e 9ef1 7fe3 888b 6a2f ae2a 352c 64e5
Package : epel-release-7-11.noarch (amazon2extra-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