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-east1b**" 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

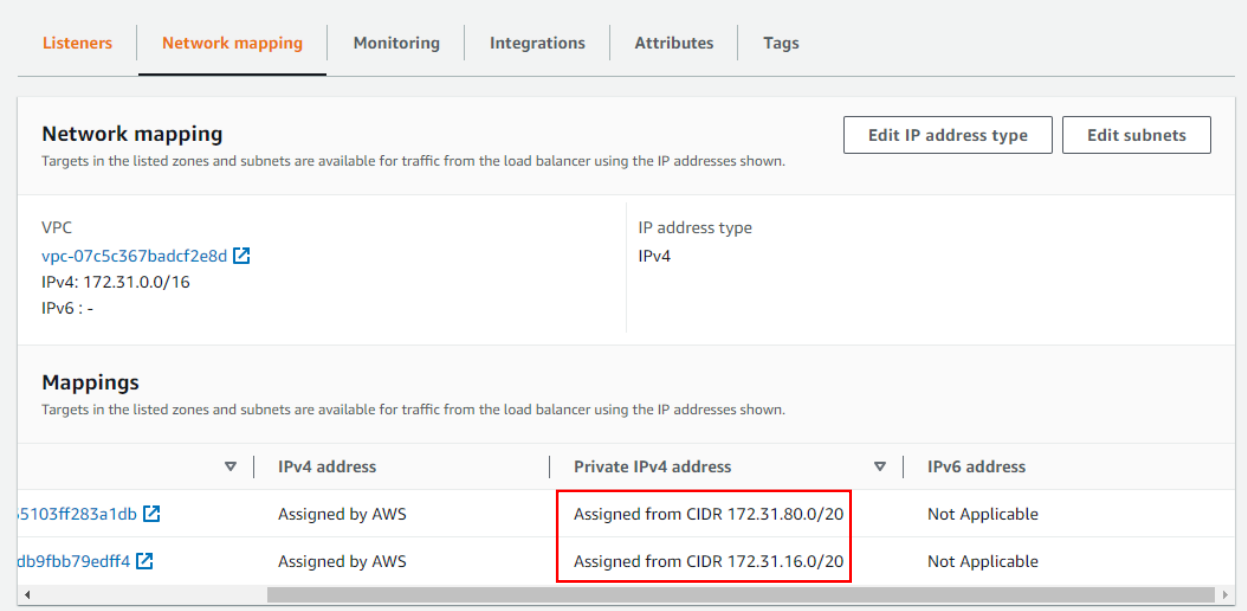
NLB takes 5 minutes to be provisioned. So please wait patiently.

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.



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.

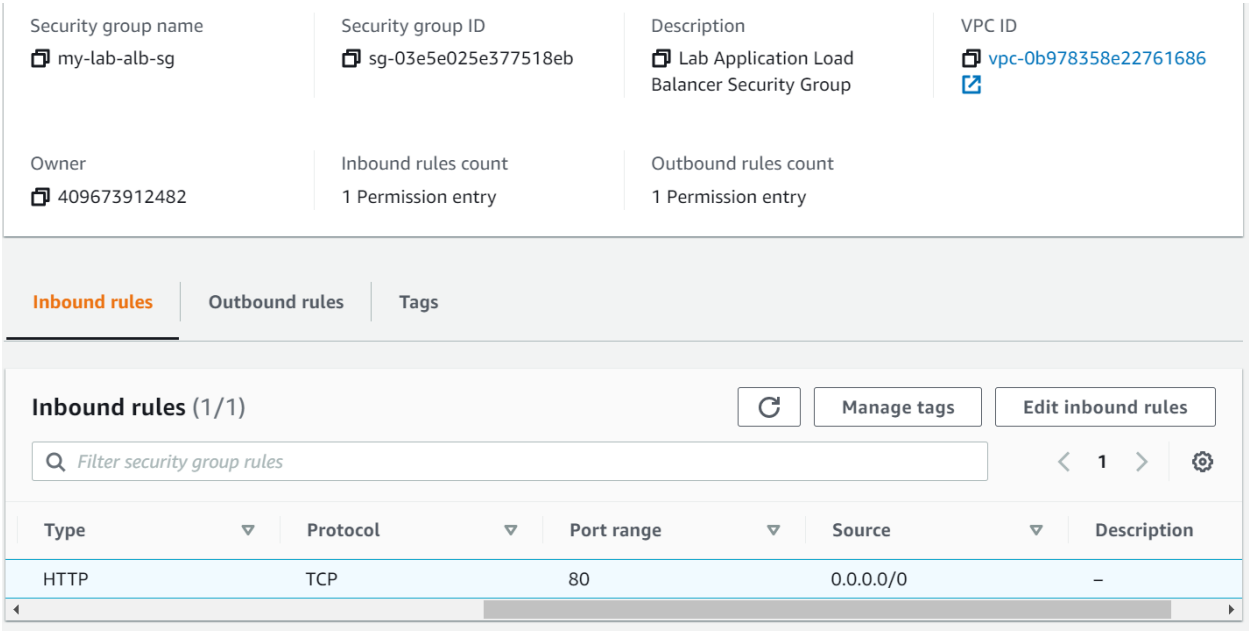
Step by step

Task 1. Run 2 web servers behind ALB

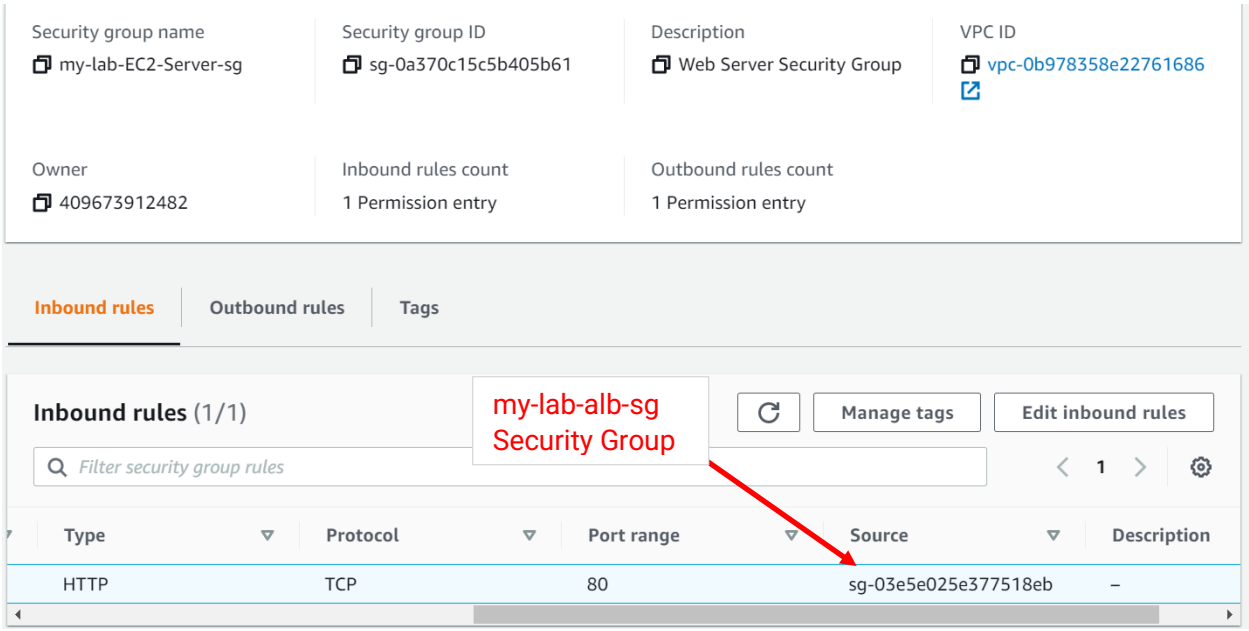
Create Security Groups for ALB

- 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)

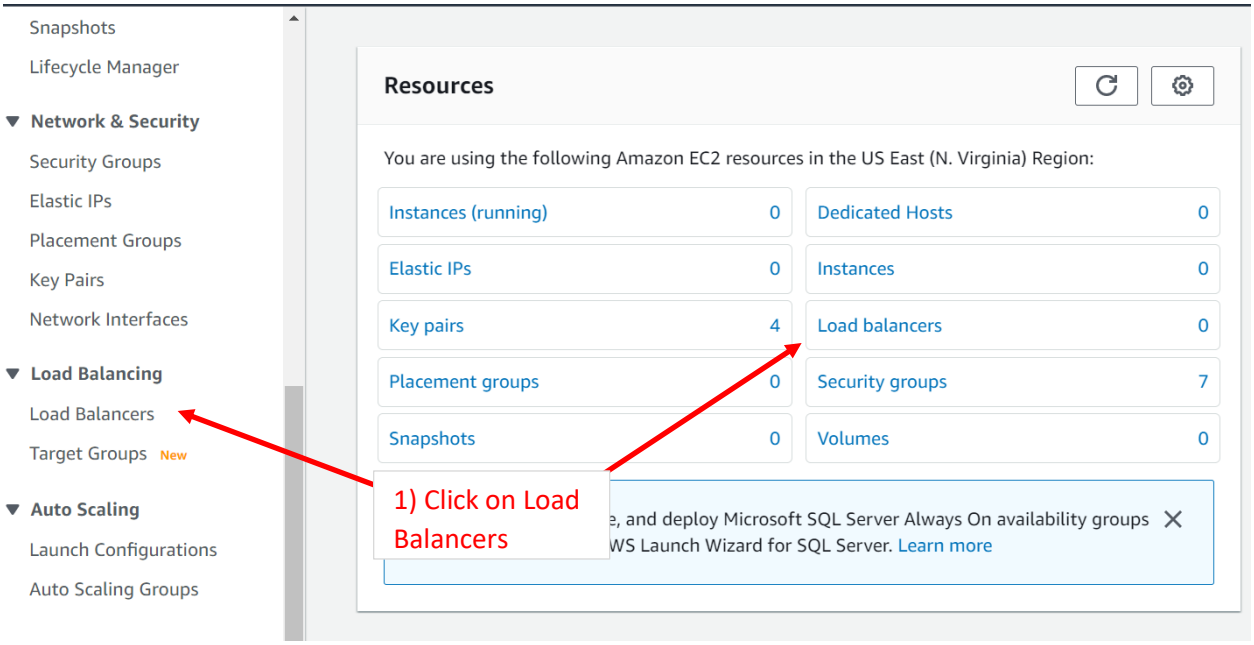


Create EC2 Web Server Security Group (Outbound Rule is Default - All Traffic)

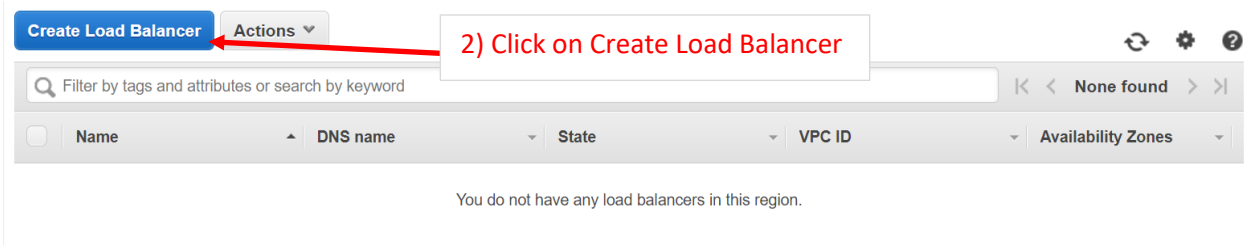


Create an ALB

Go to the Load Balancers Display from the EC2 Dashboard



...

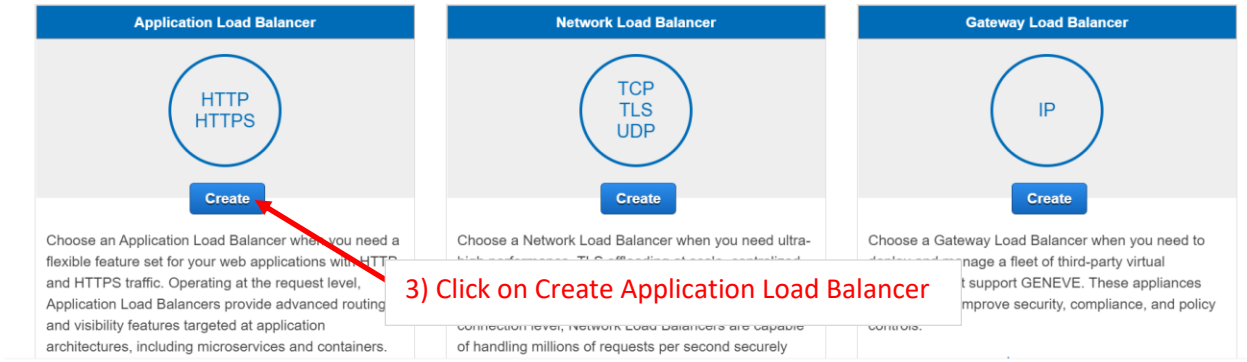


...

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 balanc type that meets your needs.

Learn more about which load balancer is right for you



3) Click on Create Application Load Balancer

...

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: my-lab-alb

Scheme: internet-facing / internal

IP address type: ipv4

4) Name Load Balancer

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 (10.0.0.0/16) | my-lab-vpc

5) Select VPC

Availability Zones:

- us-east-1a: subnet-0ef43ef1cfcb561a0 (lab-sn-public-1A) — IPv4 address: Assigned by AWS
- us-east-1b: subnet-03b7f8298553c4646 (lab-sn-public-1B) — IPv4 address: Assigned by AWS
- us-east-1c: subnet-028d2f8b4b0b12258 (lab-sn-public-1c)

6) Select At Least 2 AZ Zones and Subnets

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: VPC security groups

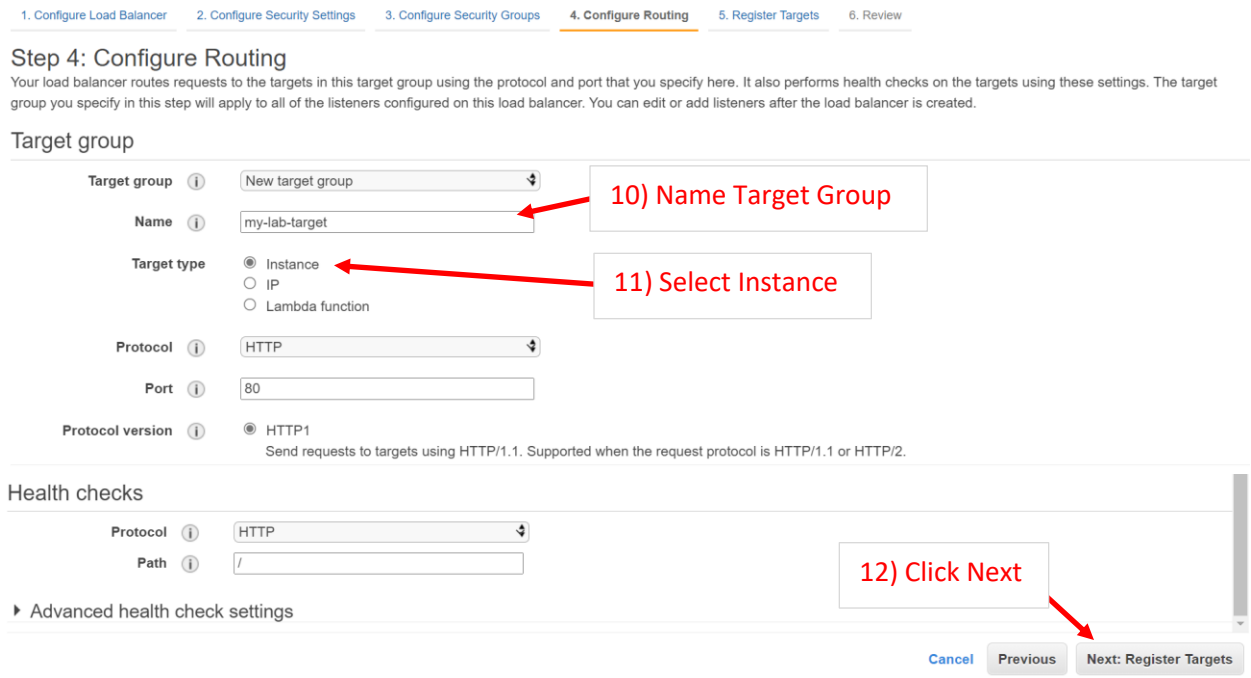
Security Group ID	Name	Description	Actions
sg-002d4b487ca1292d2	default	default VPC security group	Copy to new
sg-0d09b0bf676ce516f	launch-wizard-2	launch-wizard-2 created 2021-07-08T19:37:07.572-05:00	Copy to new
sg-0fc356187933ae278	launch-wizard-3	launch-wizard-3 created 2021-07-08T20:58:44.588-05:00	Copy to new
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

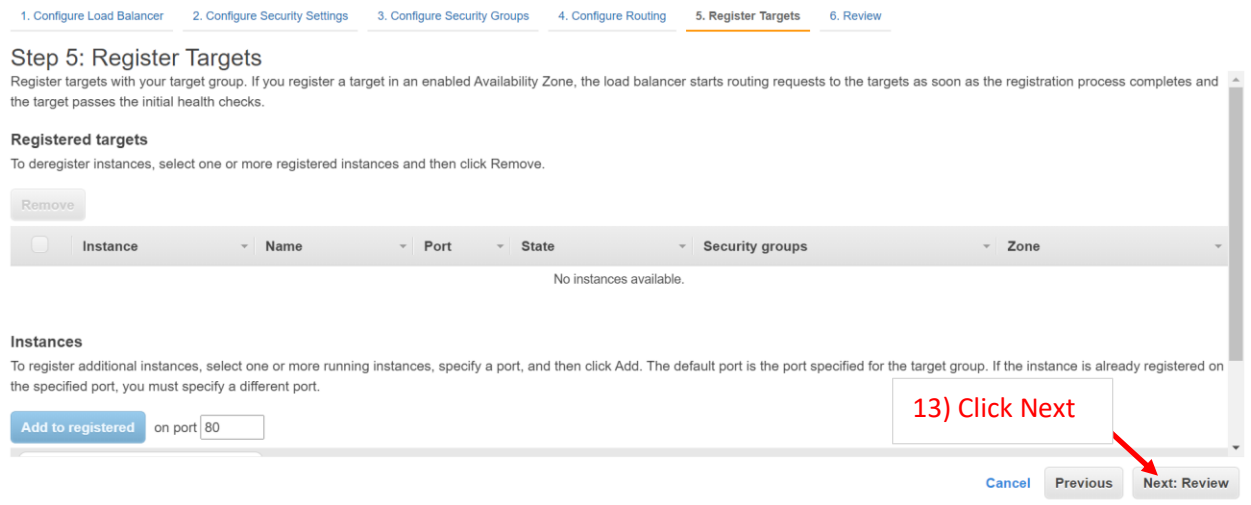
9) Click Next

Cancel | Previous | Next: Configure Routing

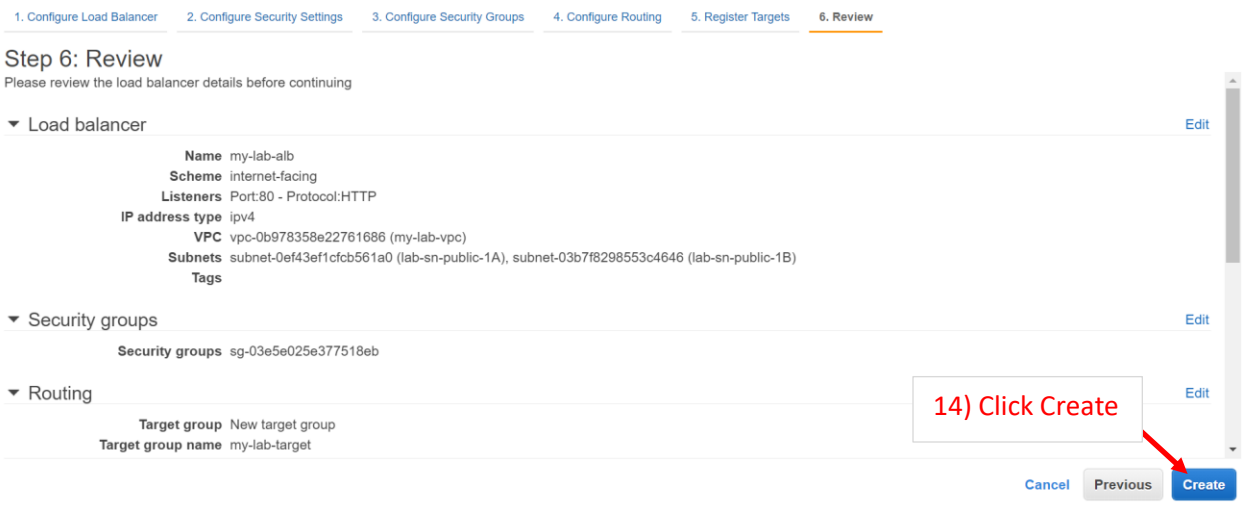
...


1. Configure Load Balancer
2. Configure Security Settings
3. Configure Security Groups
4. Configure Routing
5. Register Targets
6. Review
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
New target group
Name
my-lab-target
10) Name Target Group
Target type
Instance
IP
Lambda function
11) Select Instance
Protocol
HTTP
Port
80
Protocol version
HTTP1
Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.
Health checks
Protocol
HTTP
Path
/
12) Click Next
Advanced health check settings
Cancel
Previous
Next: Register Targets

...


1. Configure Load Balancer
2. Configure Security Settings
3. Configure Security Groups
4. Configure Routing
5. Register Targets
6. Review
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.
Add to registered
on port
80
13) Click Next
Cancel
Previous
Next: Review

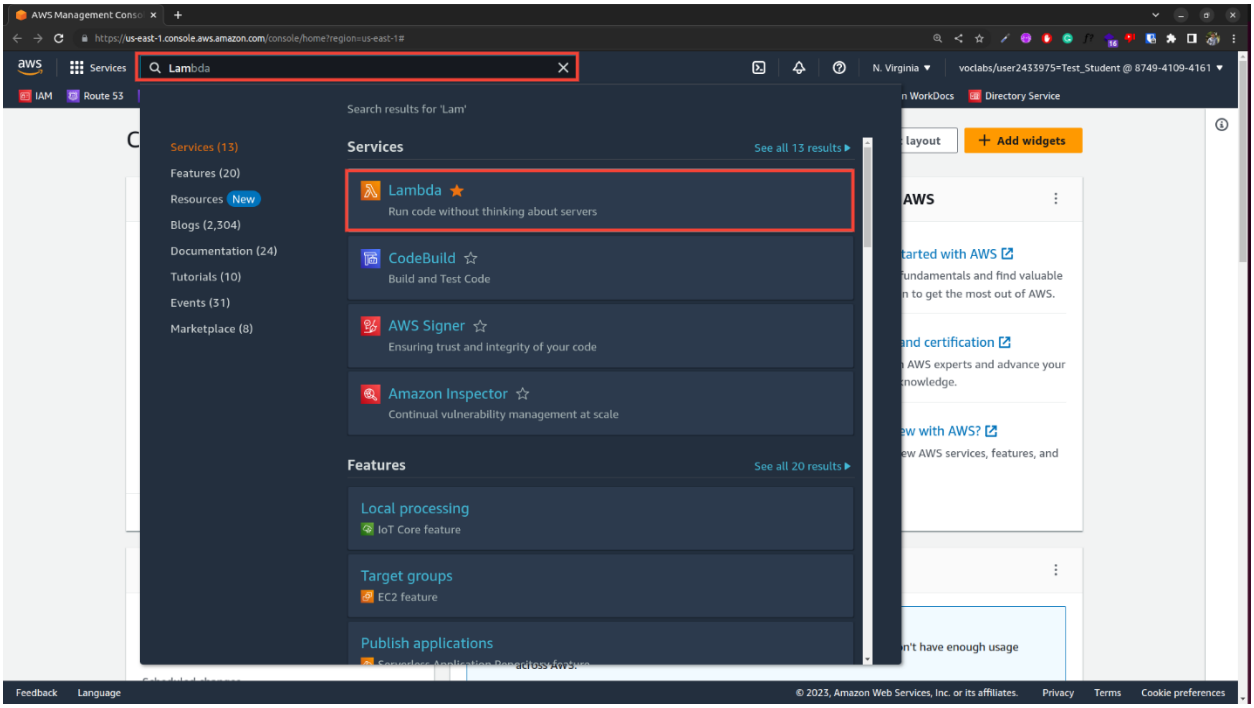
...



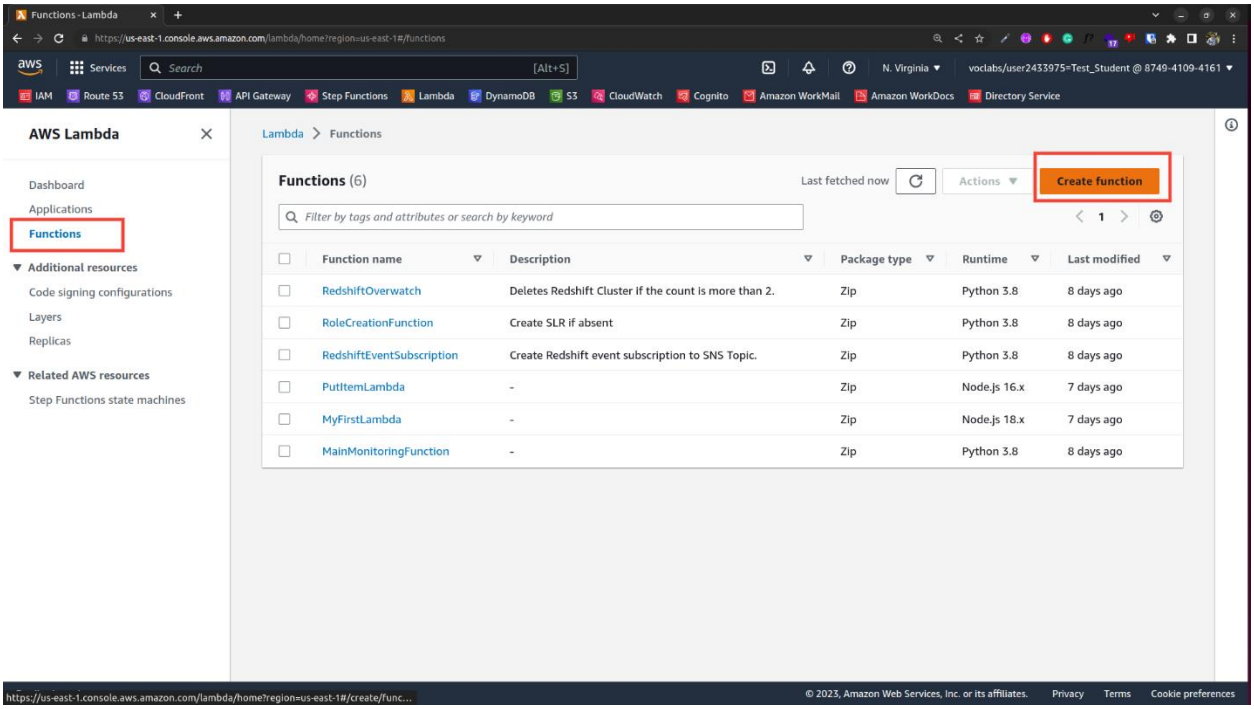
Task 2. Practice Listener rules with 2 lambdas

Part-1: Creating Lambda Function

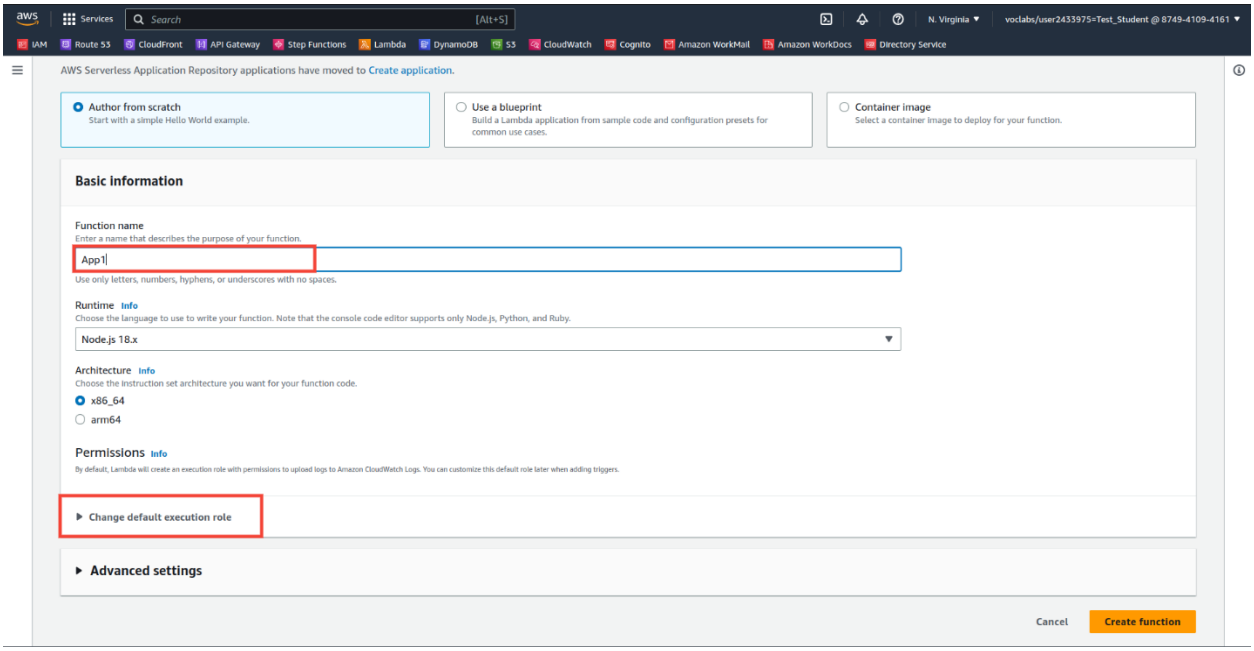
1.a Search for **Lambda** and open it.



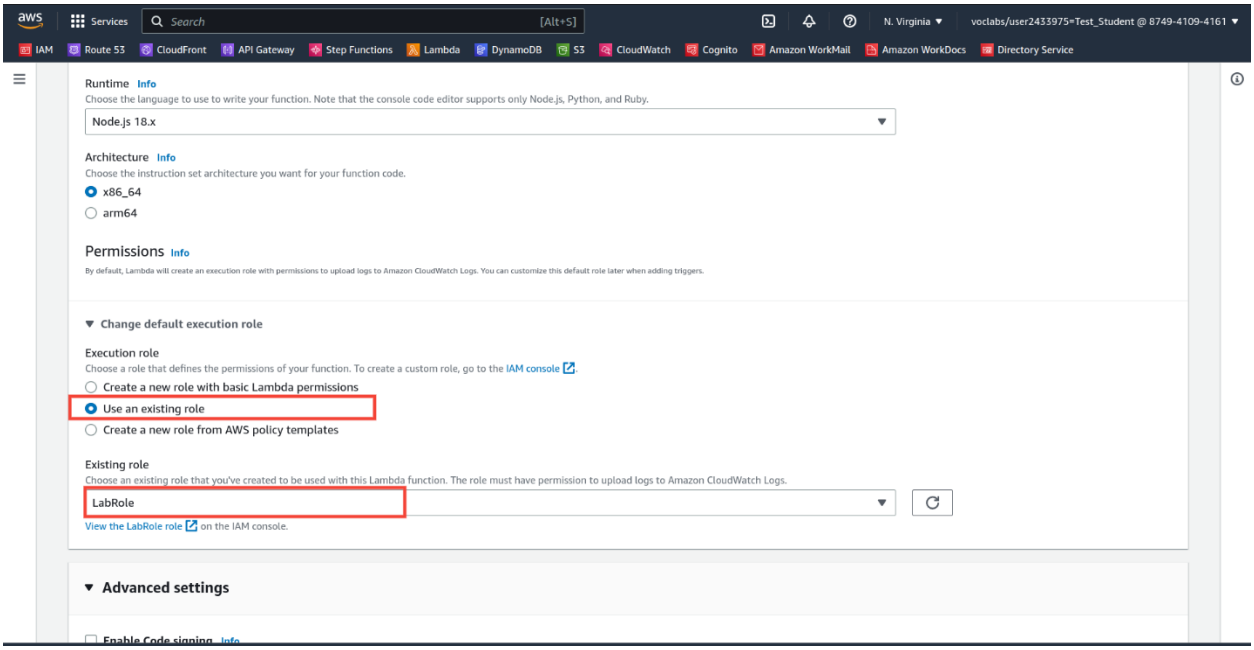
1.b In the navigation pane, choose **Functions** and then choose **Create function**.



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

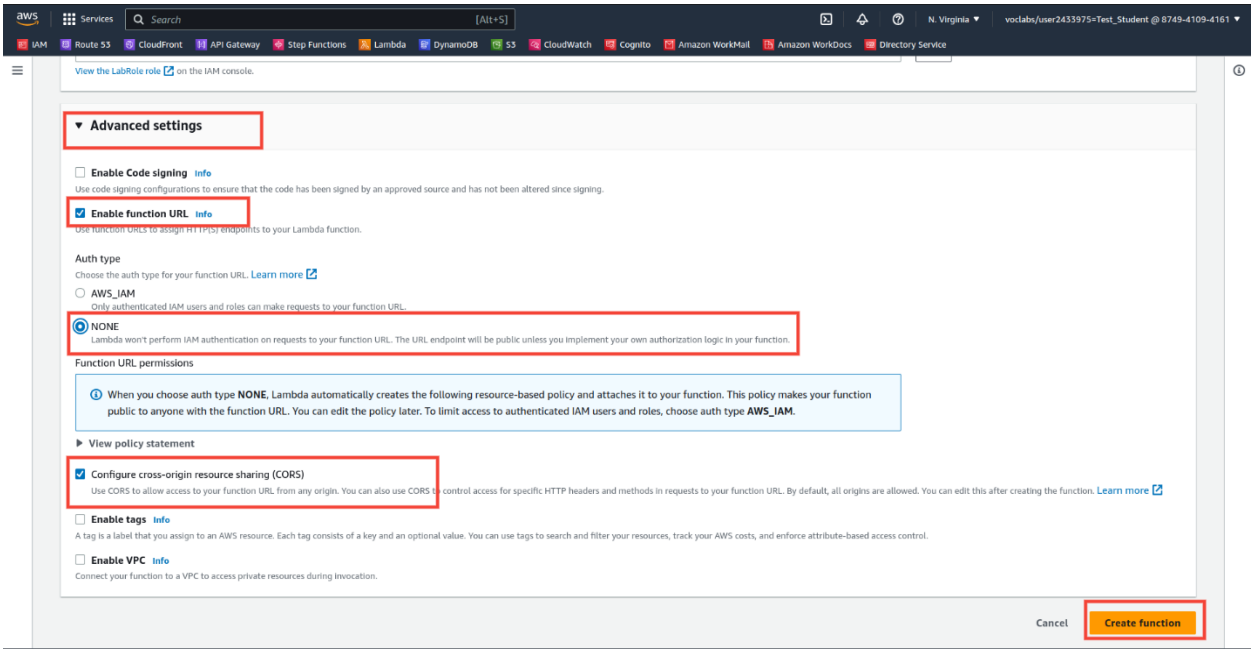


1.d Choose "**Use and Existing Role**" and select **LabRole** from the dropdown menu of Existing role.



1.e Click on Advance Settings.

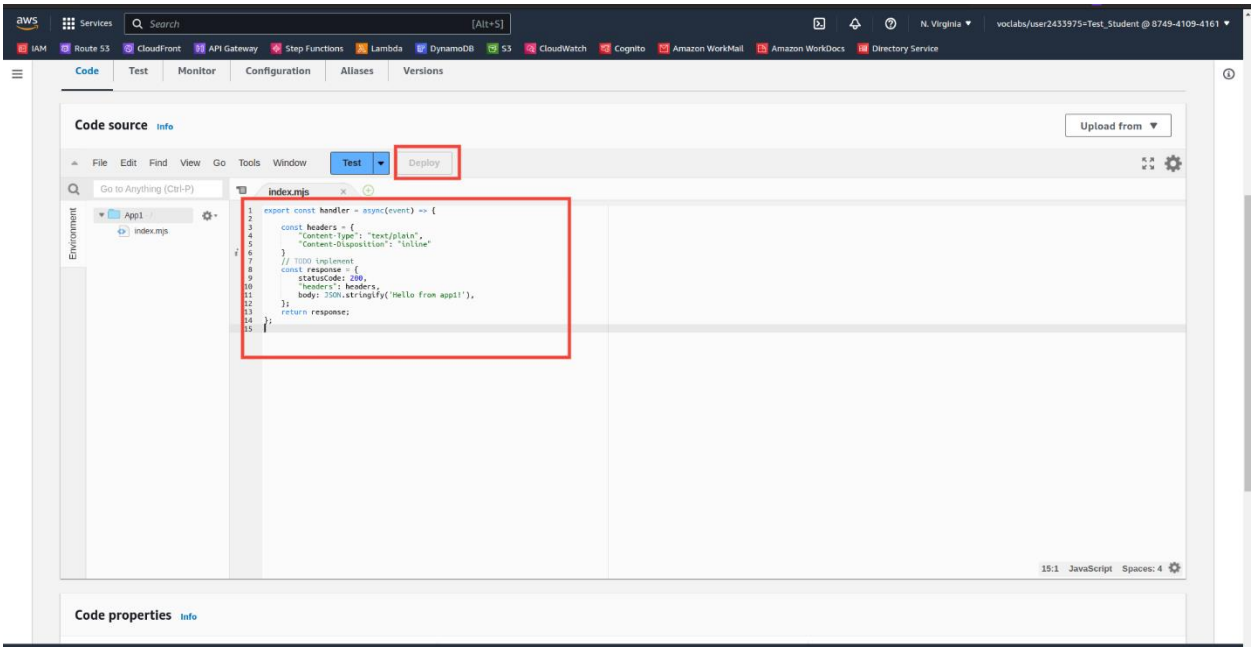
a. Select **Enable Function URL**
b. On Auth Type Select **None**
c. Select **Configure cross-origin resource sharing (CORS)**
d. Click on **Create Function**



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;
};
```

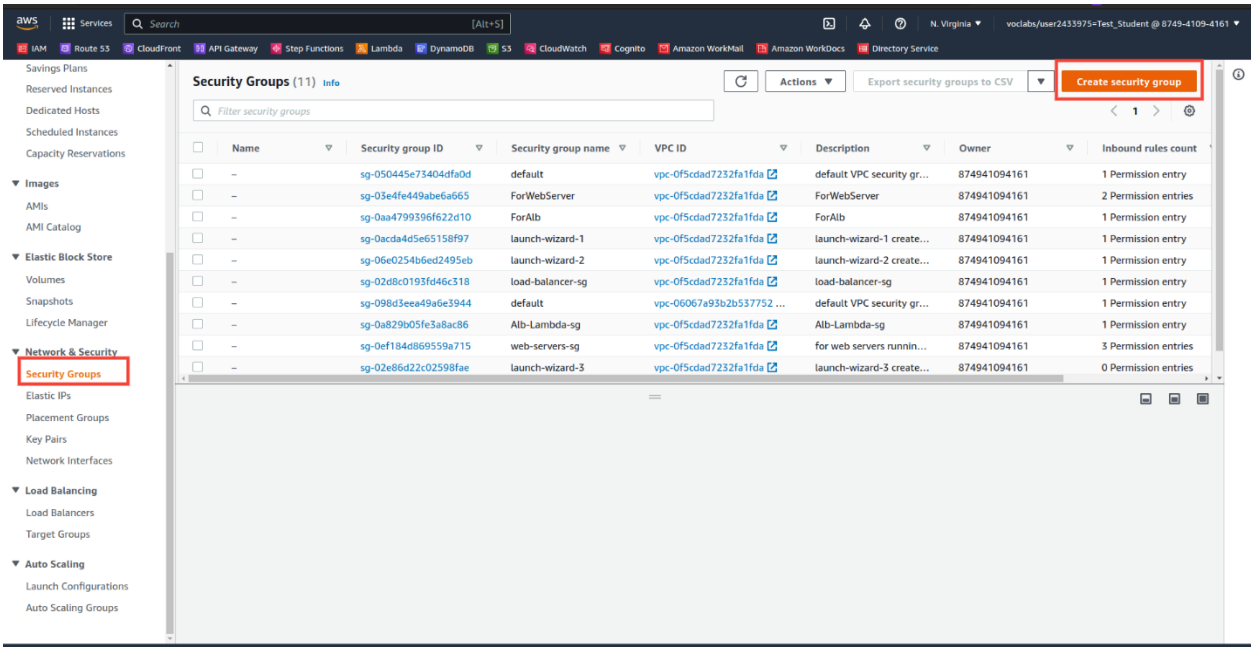


Create another Lambda with a different name. Follow the same steps to create Lambda.

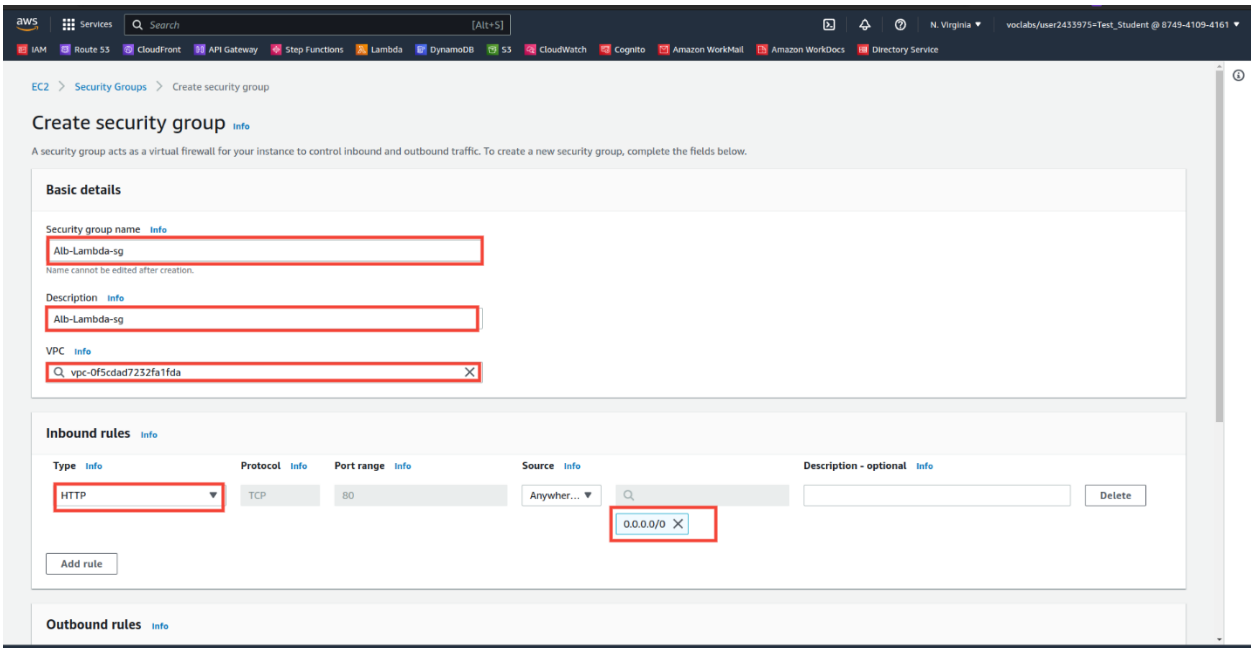
Part-2: Create Security Group for ALB.

2. a Go to EC2 Console.

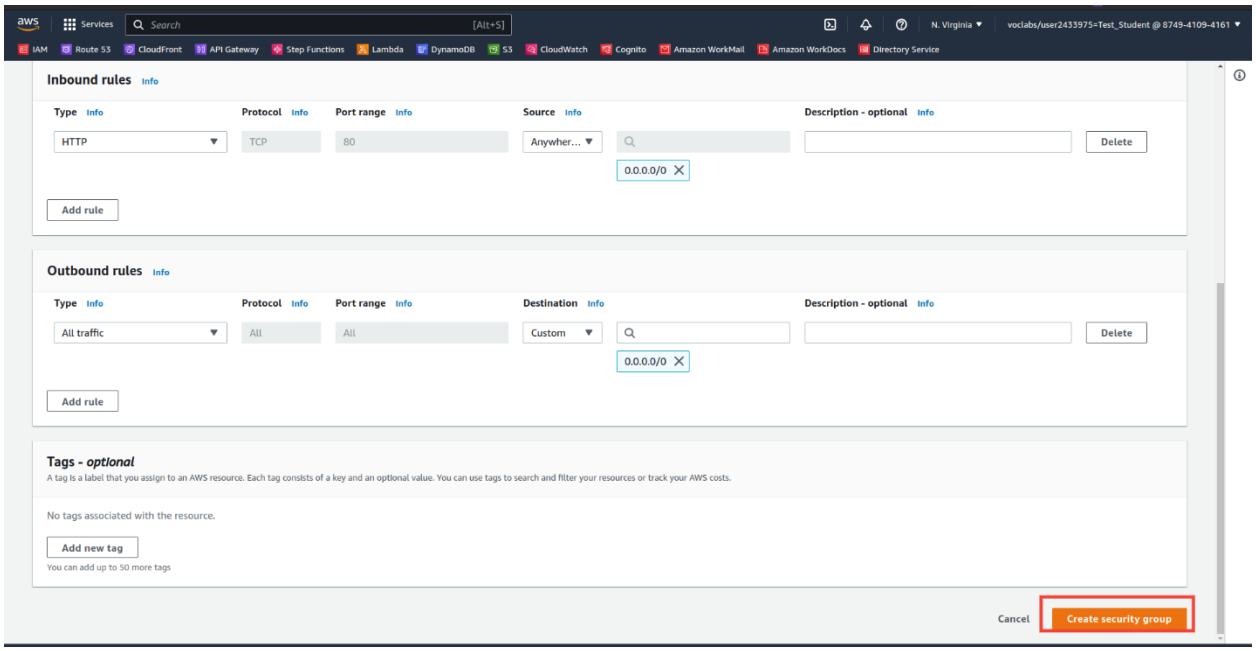
Select Security Groups from Side Bar and Click on Create Security Group.



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.

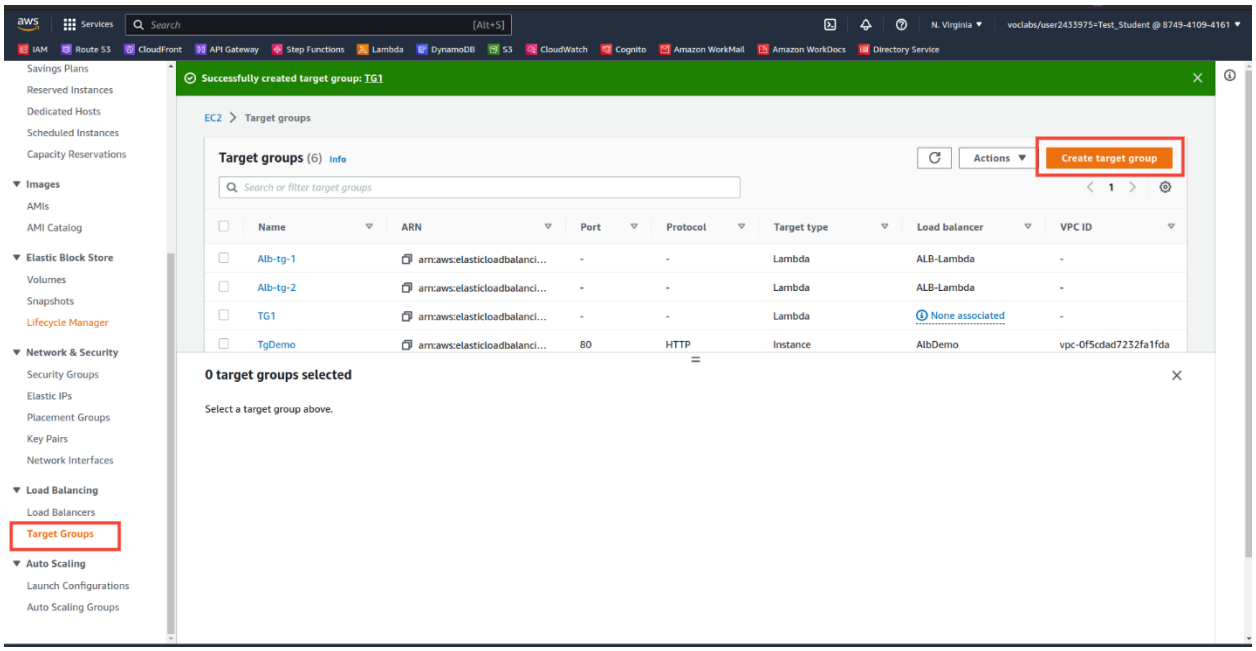


2. c Click on Create Security Group.

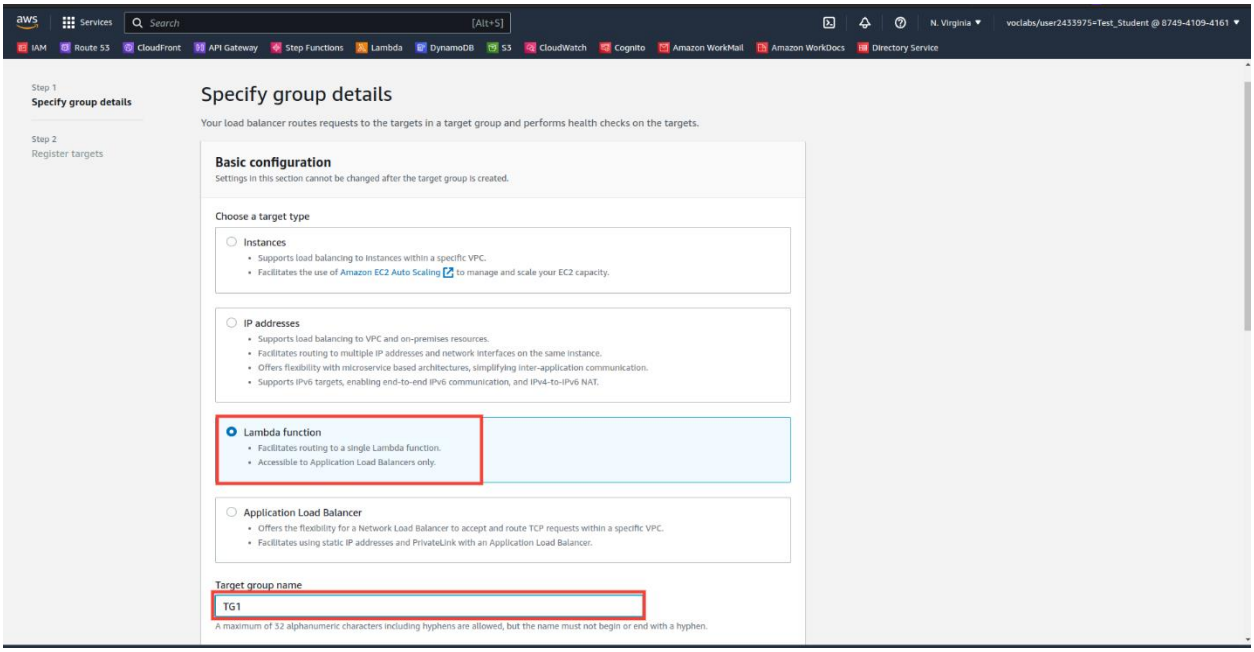


Part-3: Create Target Group TG1 and TG2.

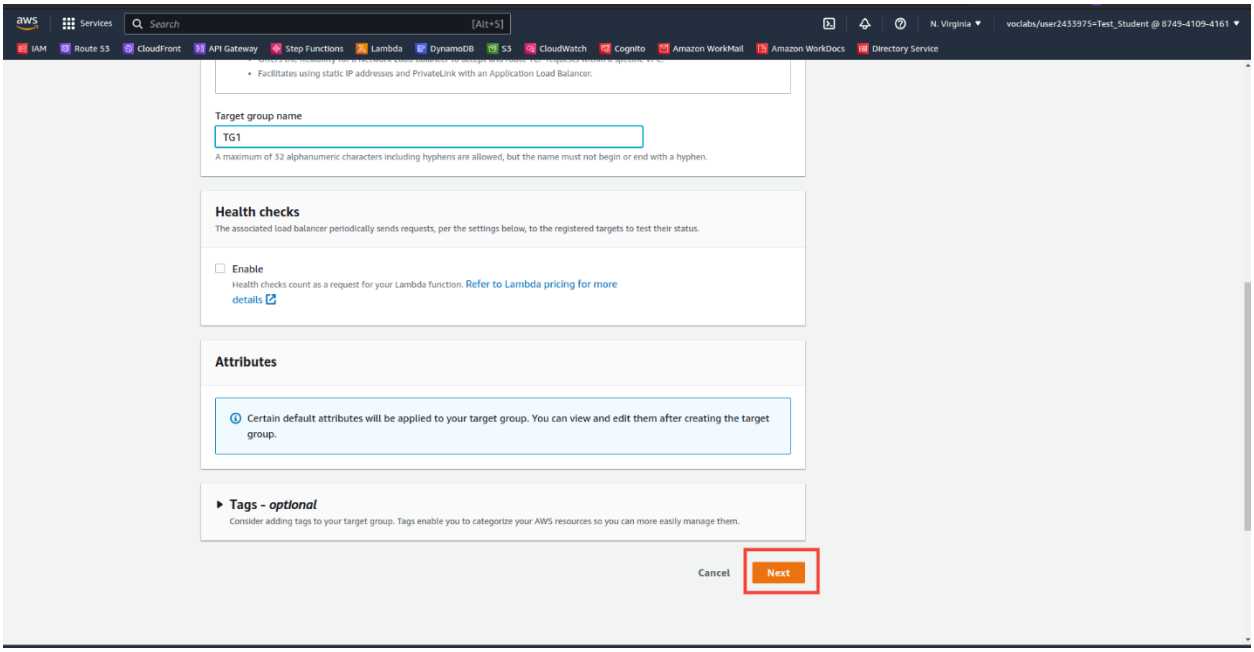
3. a Select Security Groups from Side Bar and Click on Create Security Group.



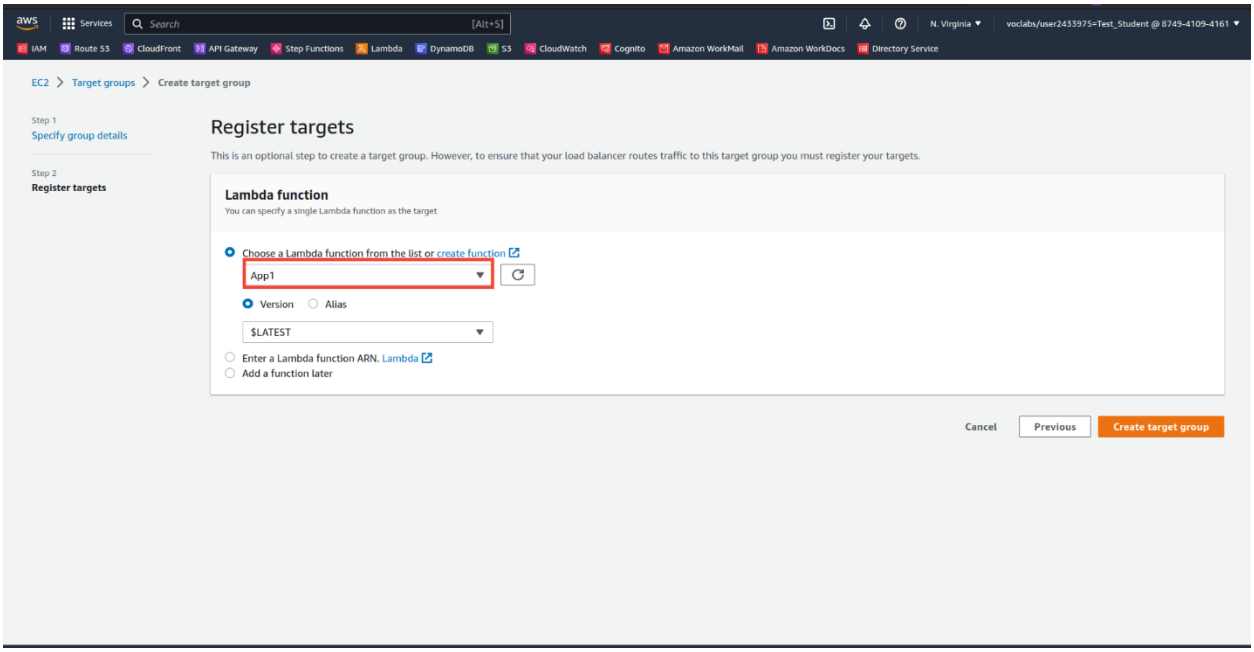
3. b Select Lambda function and fill out the Target group name.



3.c Click Next.



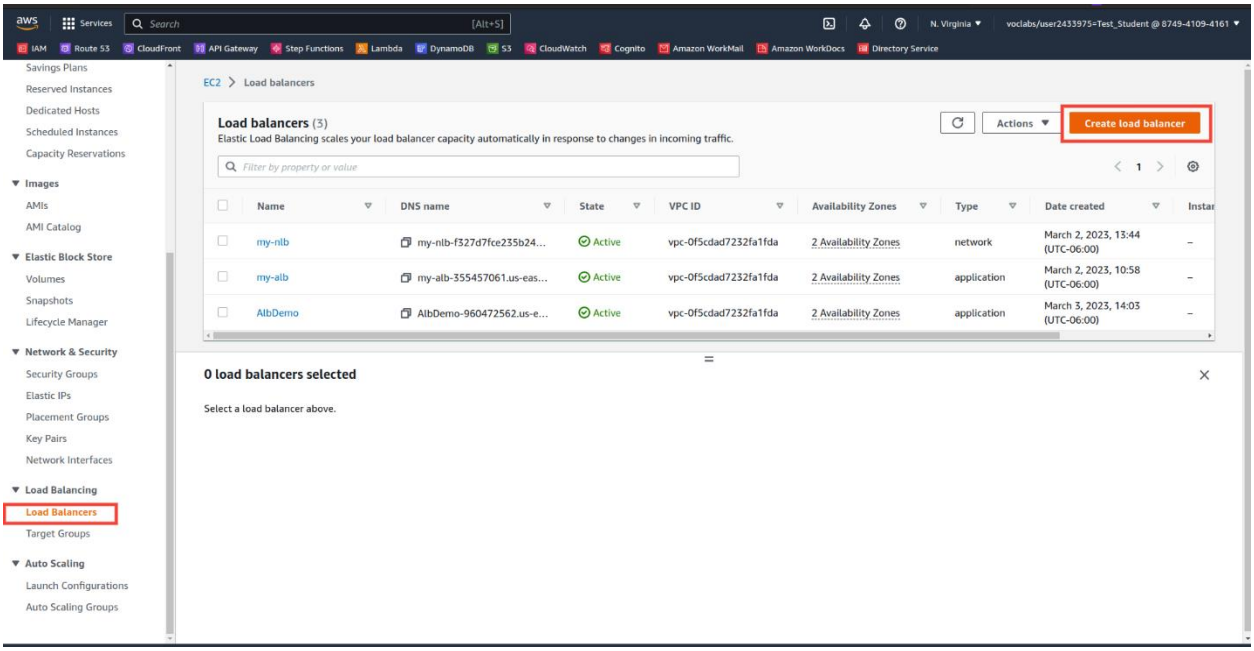
3.d Now Select the Function and Click on Create Target Group.



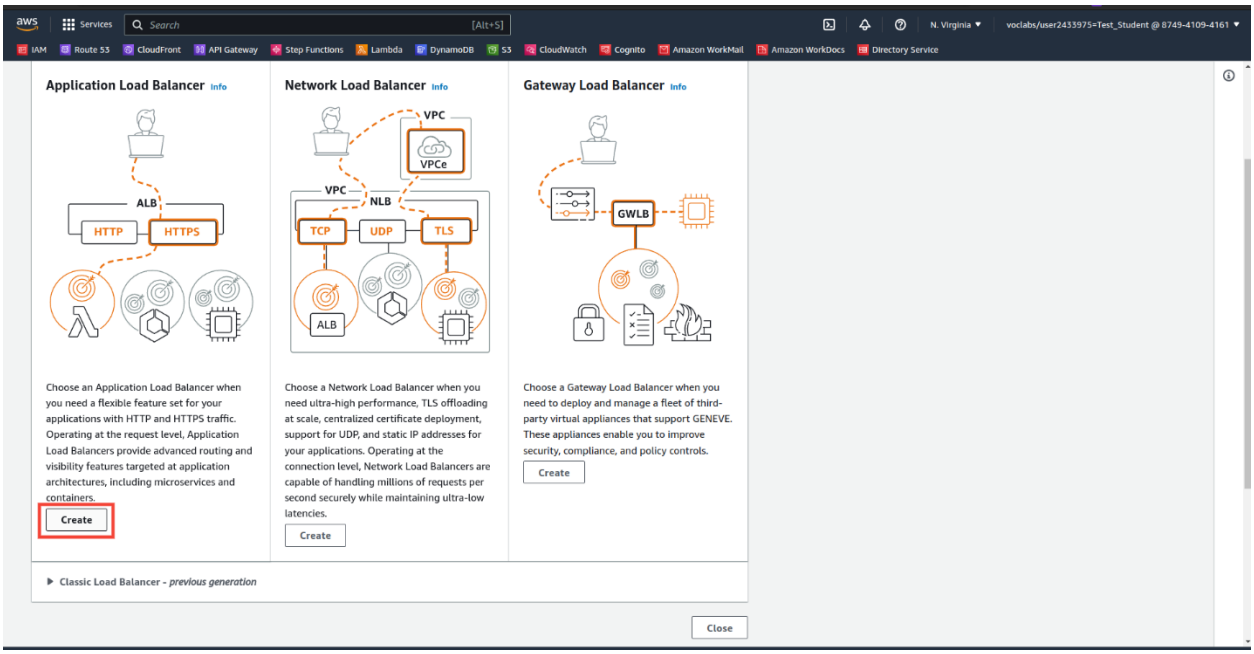
Create another Target Group with a different name. Follow the same steps to create Target Group.

Part-4: Create Application Load Balancer.

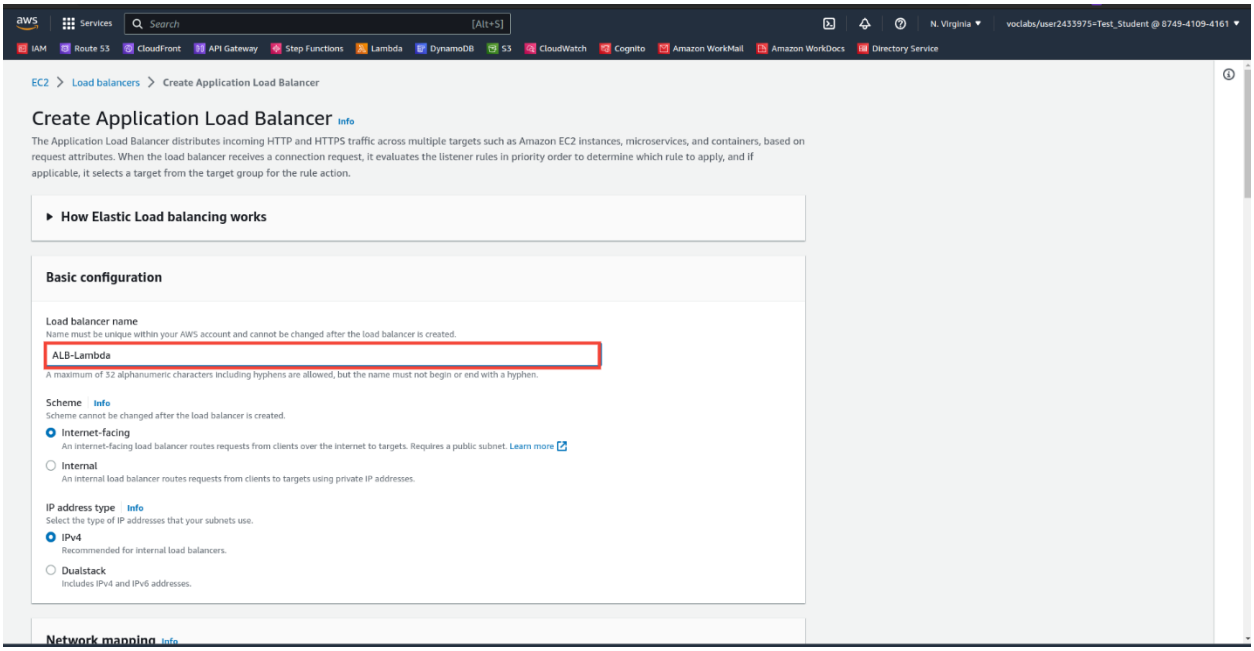
4. a Select Load Balancers from the sidebar and Click on Create load balancer.



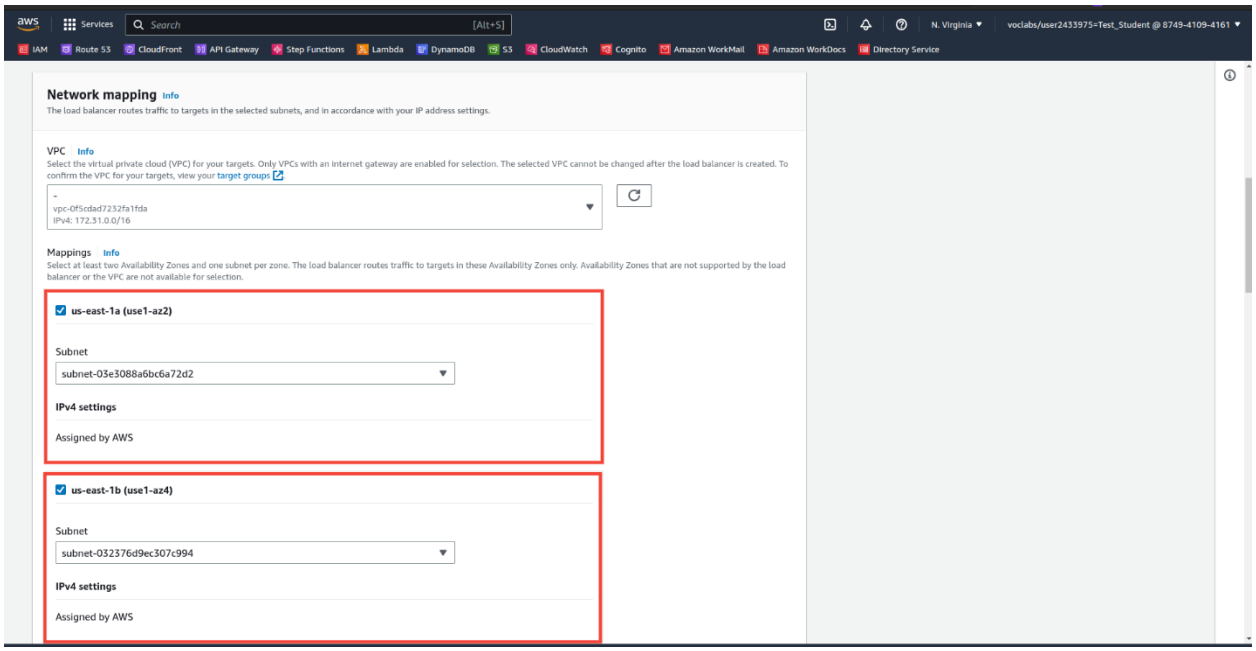
4. b Select **Create** from Application Load Balancer.



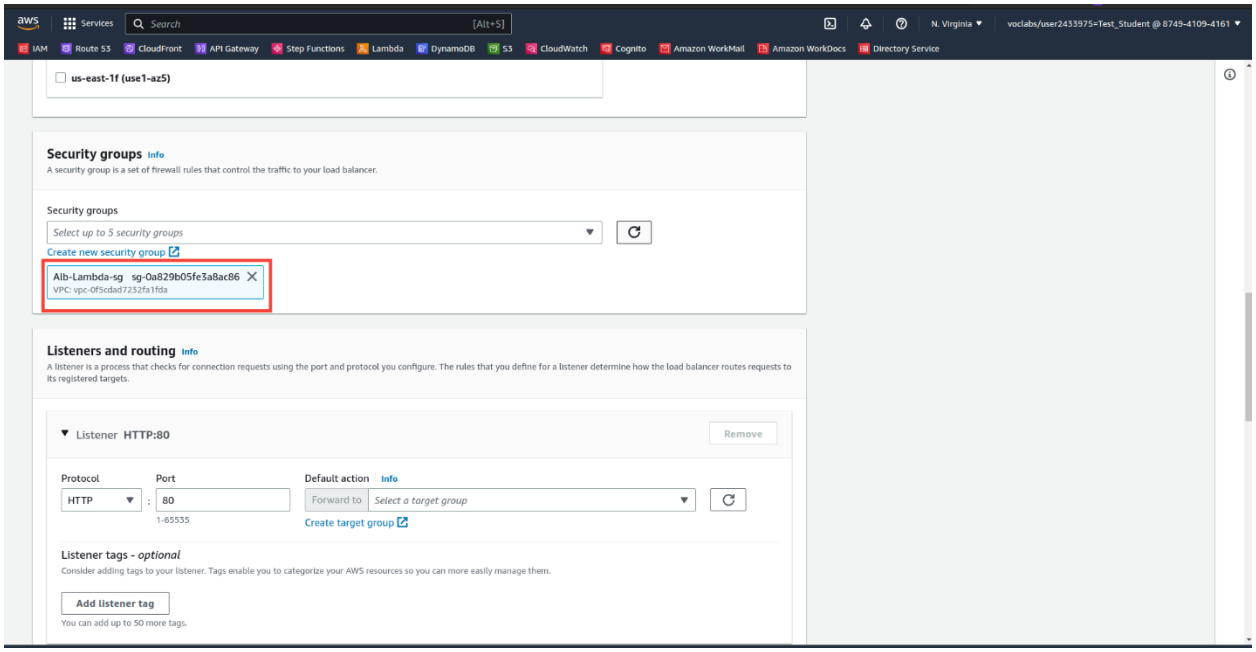
4. c Fill out the Load balancer name.



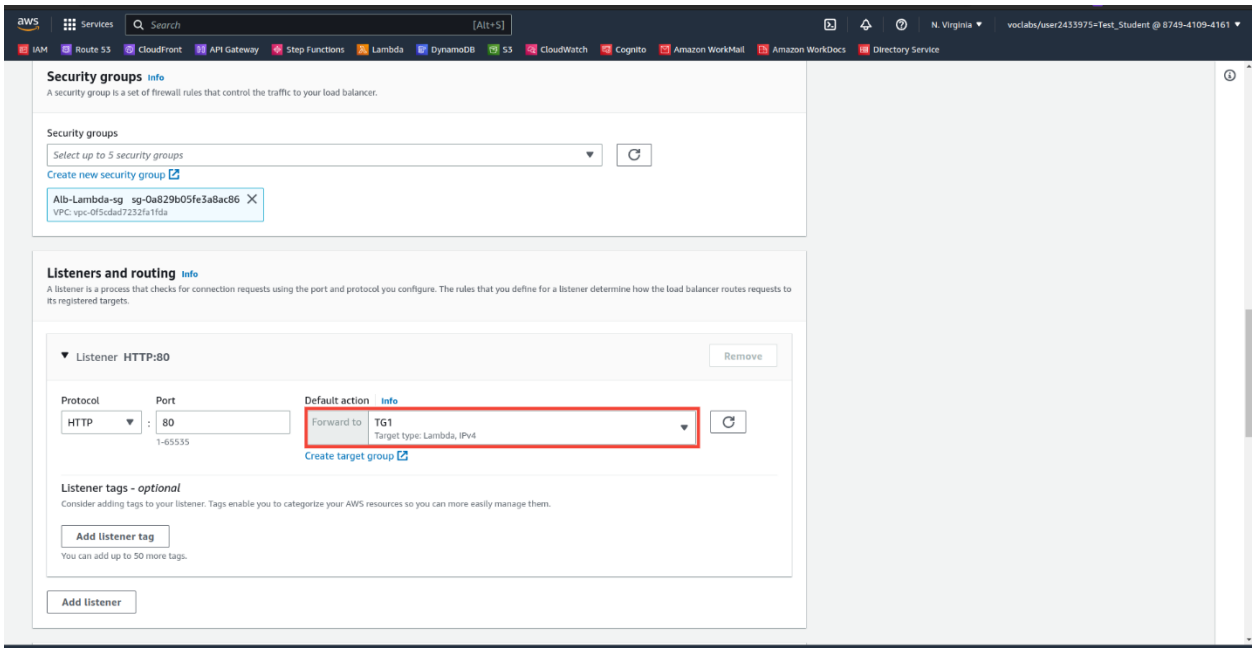
4. d Select Two availability zones. For this demo we will select us-east-1a and us-east-1b.



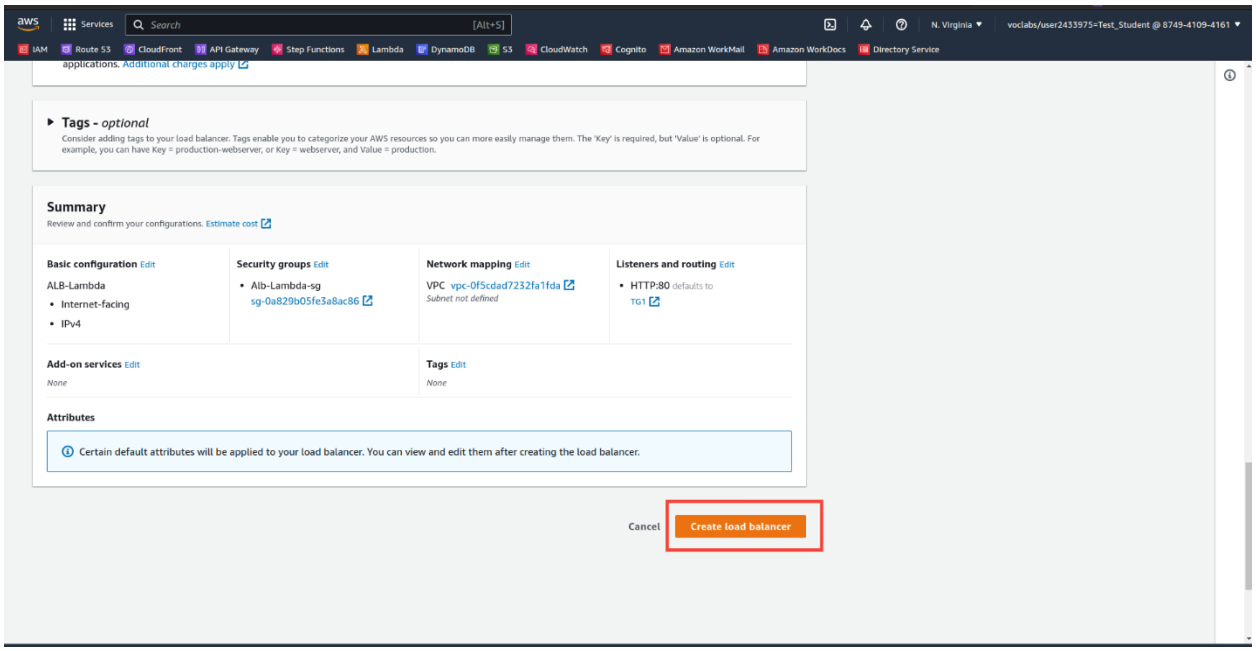
4. e Now we need to select the Security Group. Select the Security Group that we created in the previous step.



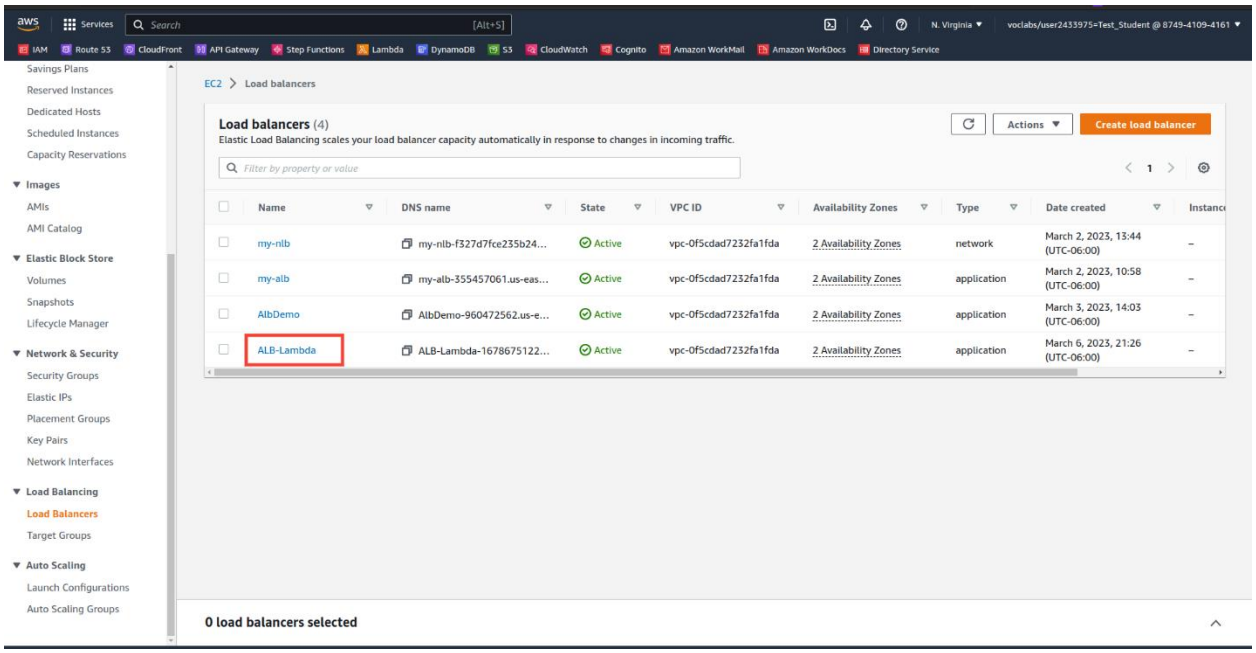
4. f Now under listeners and routing, select TG1 as the target group. We created this target group in the previous step.



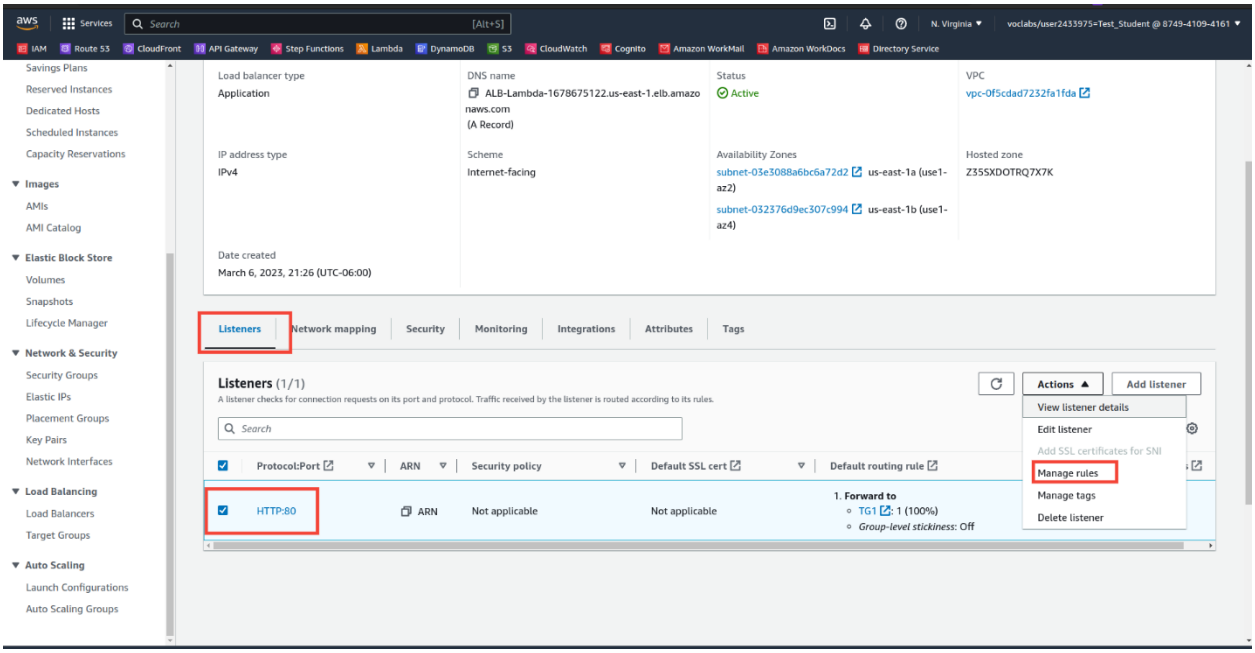
4. g Click on Create load balancer.



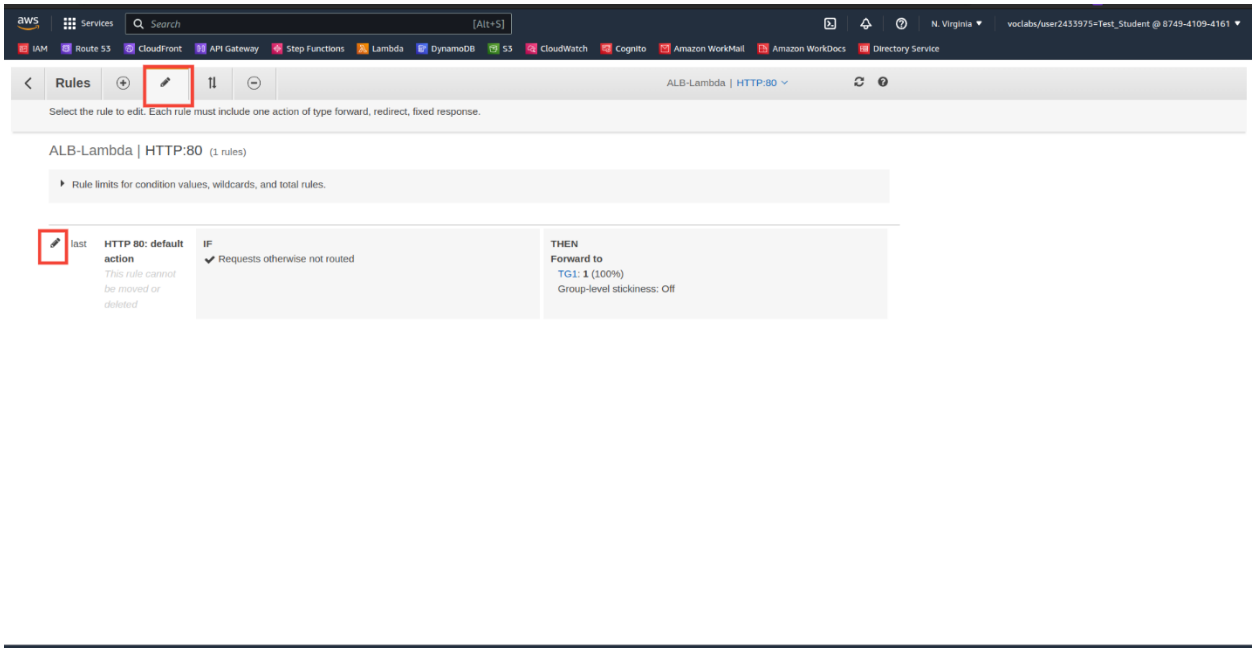
4. h Now Go to Load Balancers and Click on the load Balancer that you created.



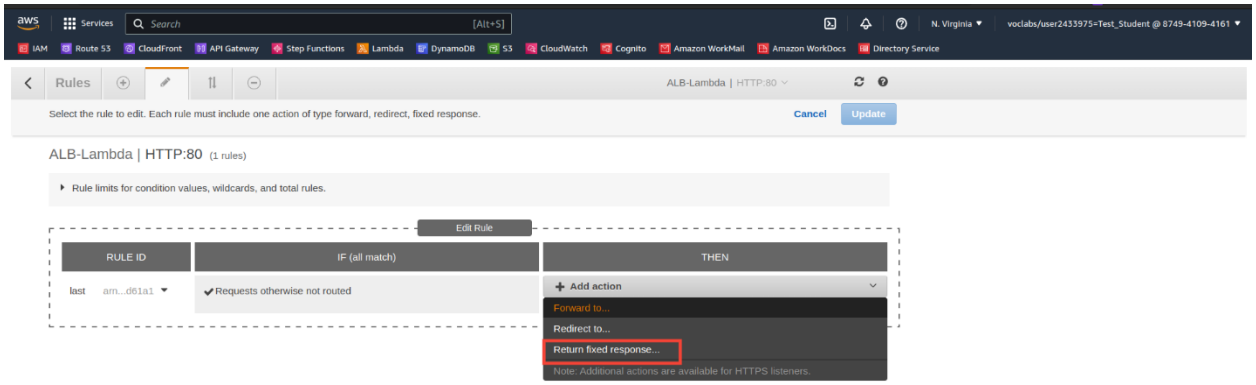
4.i Scroll down and select the protocol. After that Click on Actions and select Manage rules.



4. j Click on Edit Icon.

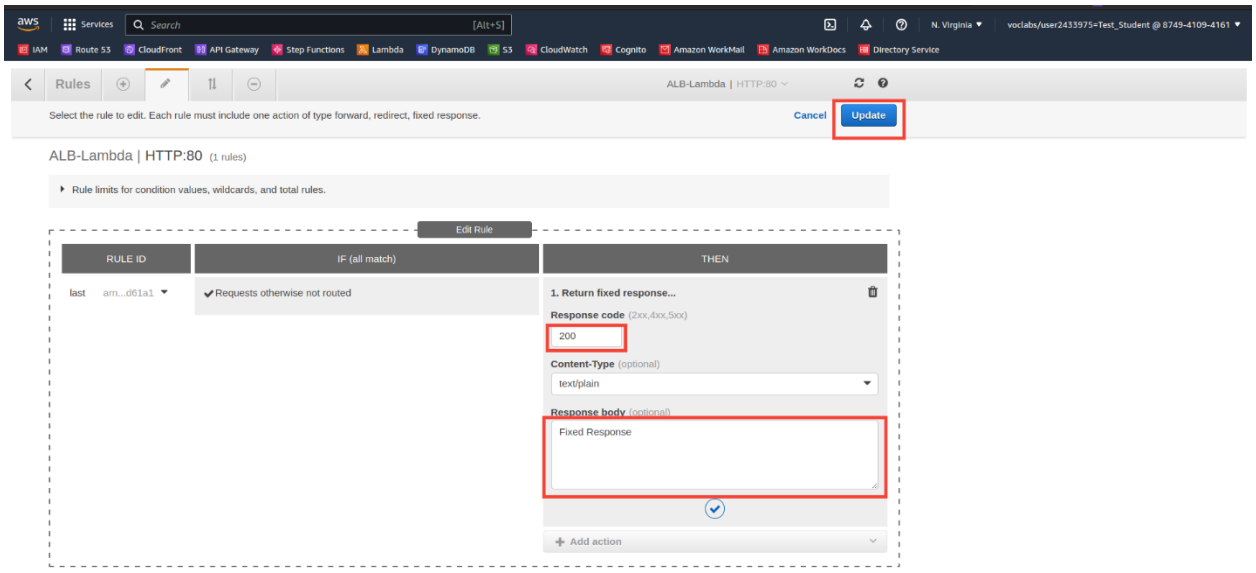


4. k Delete the THEN section and Add Return Fixed Response Action.

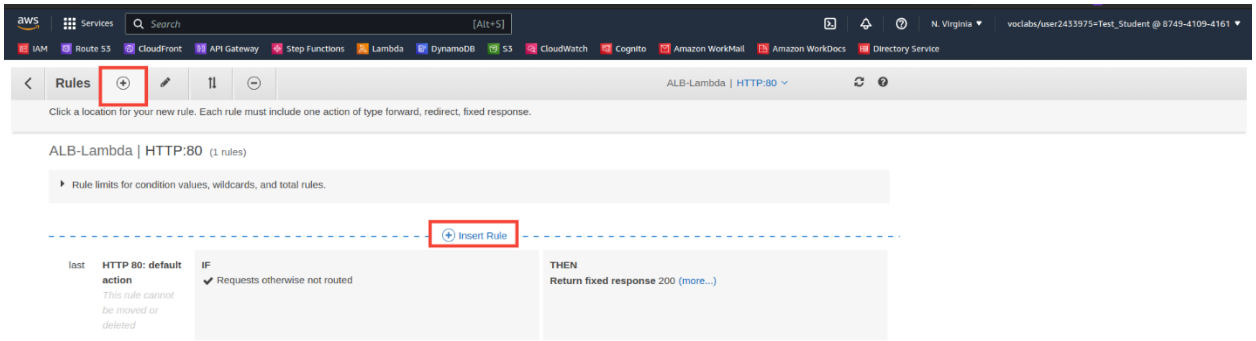


4.l

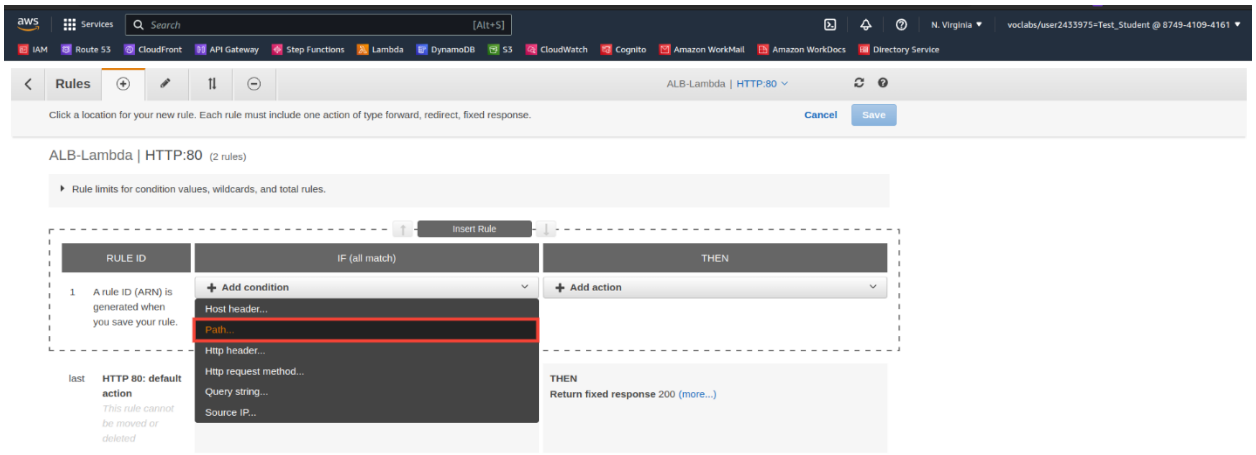
Change Response Code to 200 and Change Response Body to" Fixed Response". Lastly Click Update.



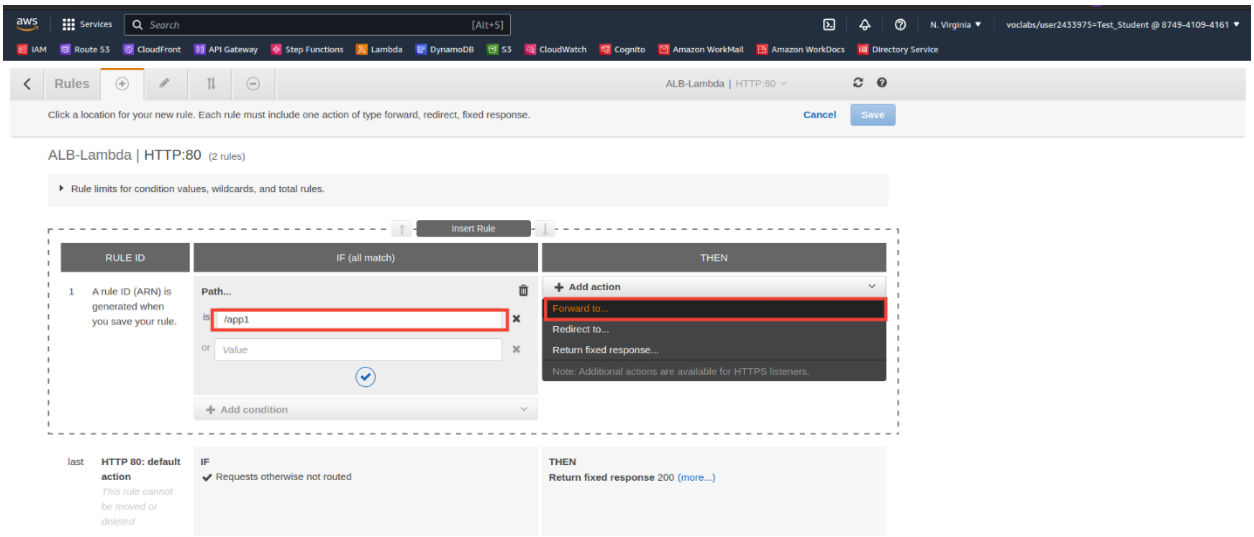
4.n Now click first Click on ⊕. Then click on Insert Rule.



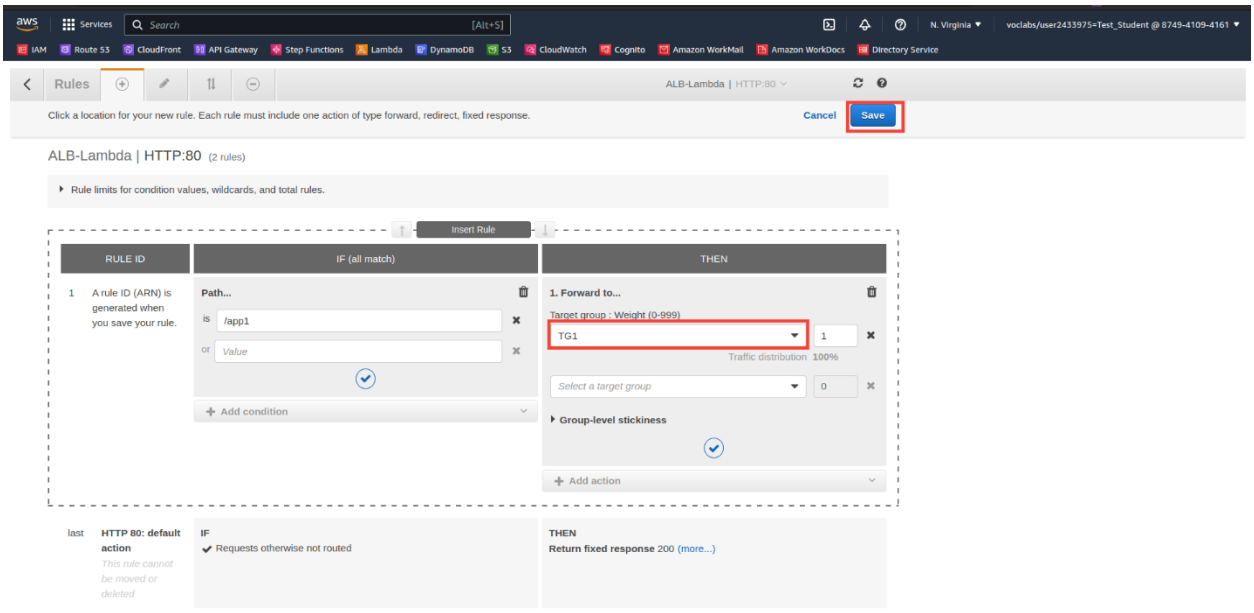
4.o Now click on Add condition and Select Path.



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



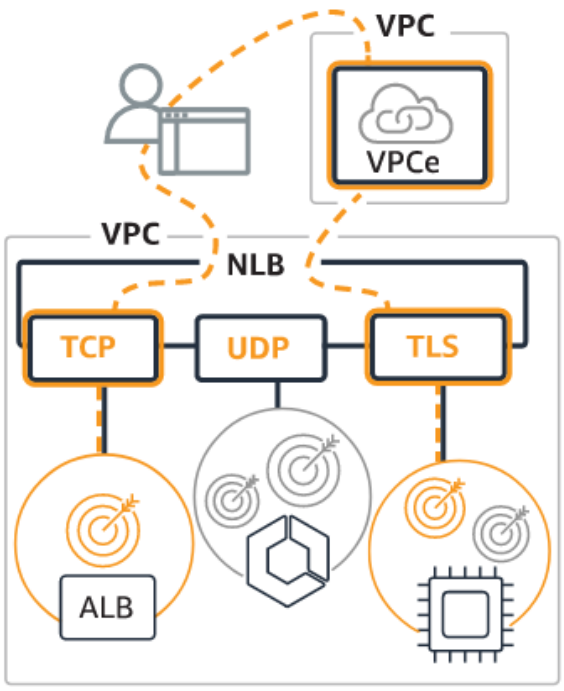
4 q. Now Select Target Group as TG1 and Click on Save.



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	t2.micro	2/2 checks passed	No alarms	us-east-1a
myNlbE1	i-03f9925915e9136a6	Running	t2.micro	Initializing	No alarms	us-east-1a
myNLBE2	i-009a5fdf725c0d8d2	Running	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.

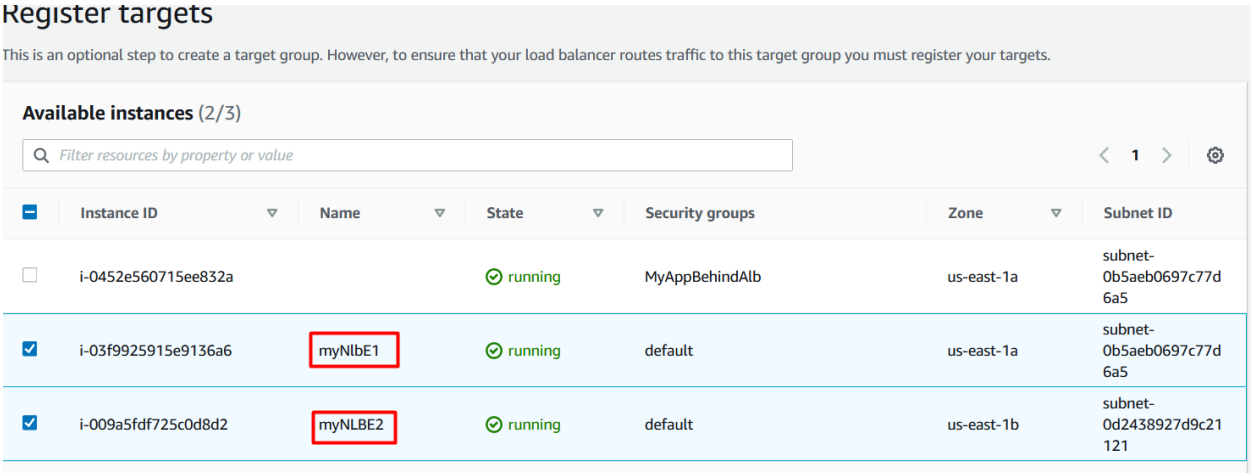
Protocol Port

TCP : 80

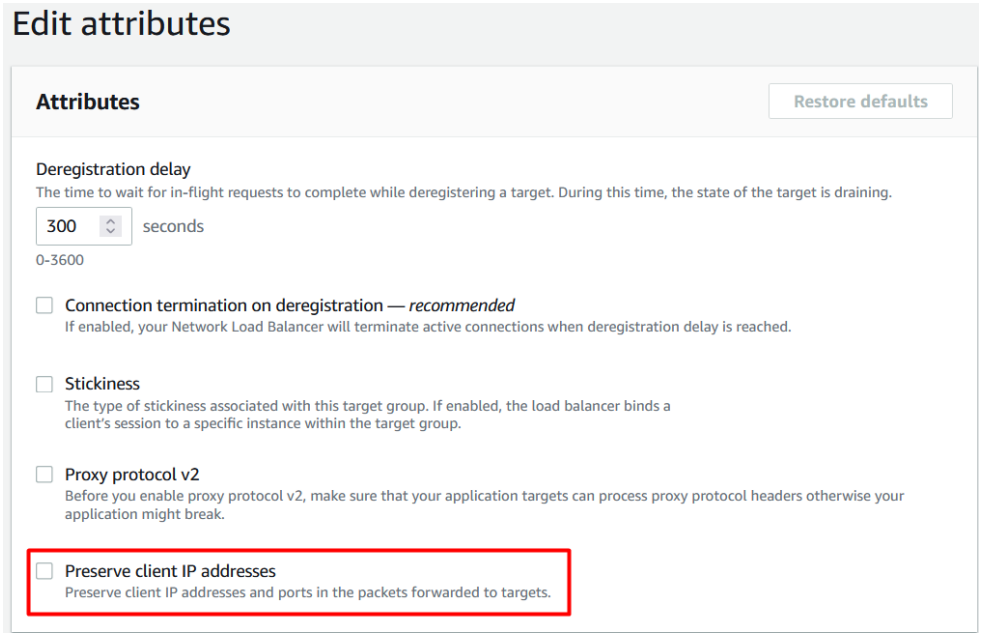
VPC

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

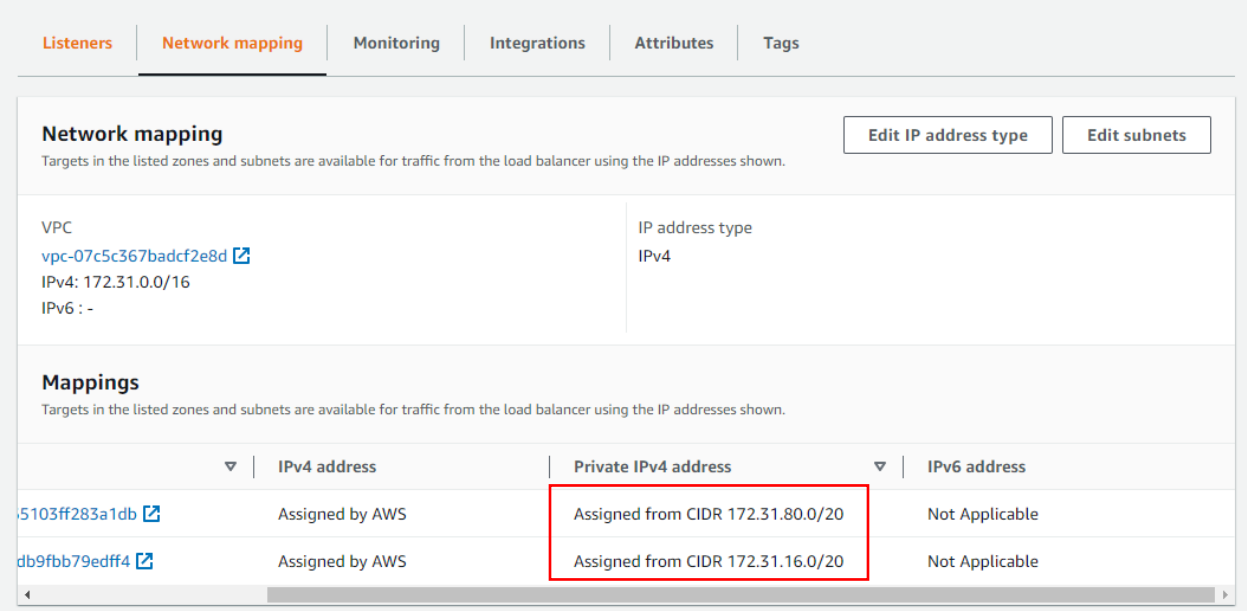
my-first-vpc
vpc-0def861cf2ef04f24
IPv4: 10.0.0.0/16

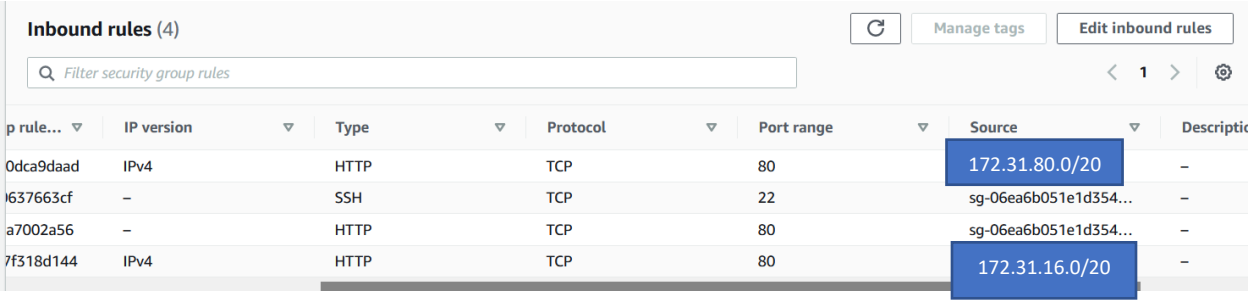


c. Update the target group and deselect Preserve client IP addresses



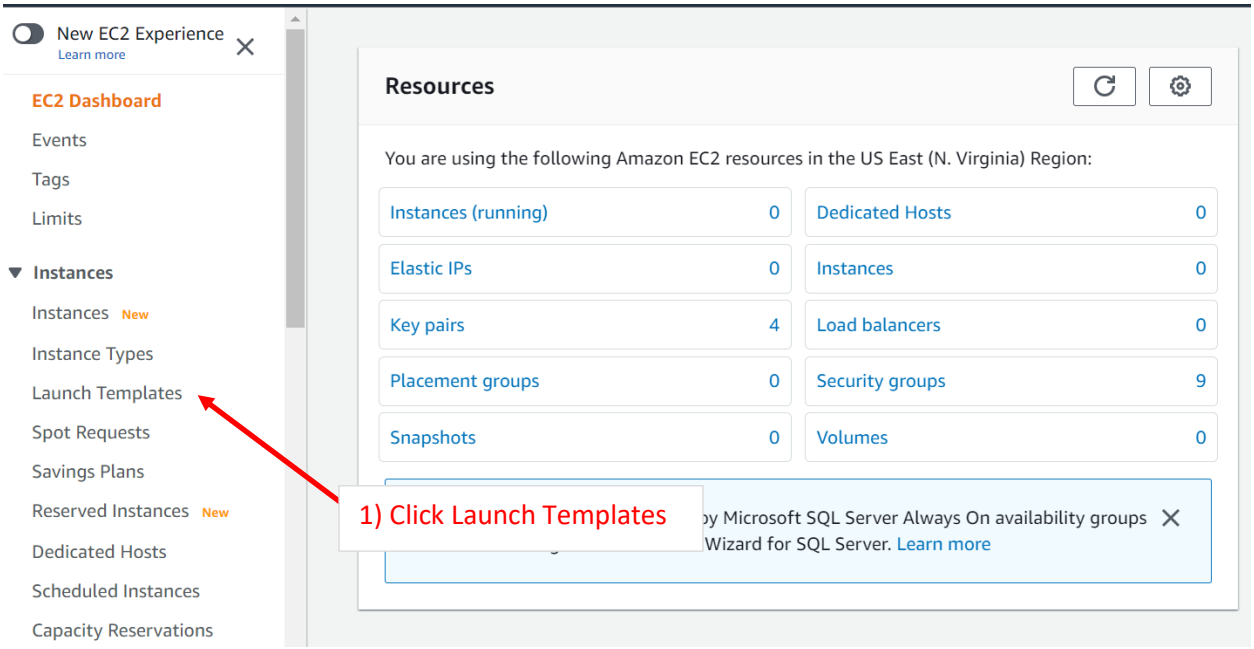
f. Grab private subnets. Update the instance's security group to allow access from the NLB nodes created in those subnets.


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
vpc-07c5c367badcf2e8d
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
Assigned by AWS
Assigned from CIDR 172.31.80.0/20
Not Applicable
db9fbb79edff4
Assigned by AWS
Assigned from CIDR 172.31.16.0/20
Not Applicable

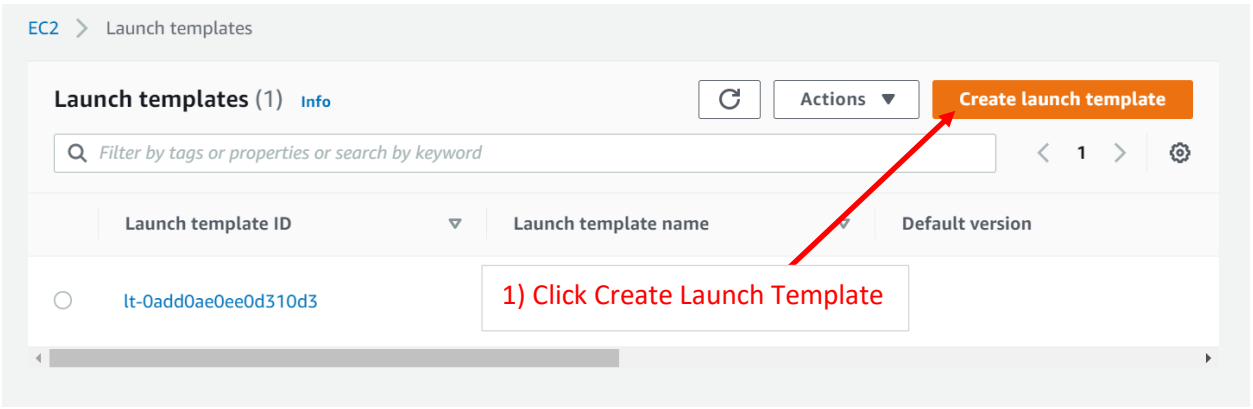

Inbound rules (4)
Manage tags
Edit inbound rules
Filter security group rules
1
p rule...
IP version
Type
Protocol
Port range
Source
Descriptio
0dca9daad
IPv4
HTTP
TCP
80
172.31.80.0/20
637663cf
–
SSH
TCP
22
sg-06ea6b051e1d354...
a7002a56
–
HTTP
TCP
80
sg-06ea6b051e1d354...
7f318d144
IPv4
HTTP
TCP
80
172.31.16.0/20

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

Go to Launch Templates Display from EC2 Display



Create Launch Template



...

EC2 > Launch templates > Create launch template

Create launch template

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

Launch template name and description

2) Name Template

Launch template name - *required*

my-lab-server

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

Template version description

A prod webserver for MyApp

Max 255 chars

Selecting Guidance will show more Details

Auto Scaling guidance **Info**

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

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

▶ **Template tags**

▶ **Source template**

Launch template contents

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

▼ Amazon machine image (AMI) - required Info

3) Select AMI

AMI - *required*

Amazon Linux 2 AMI (HVM), SSD Volume Type
ami-0dc2d3e4c0f9ebd18
Catalog: Quick Start virtualization: hvm architecture: 64-bit (x86)

▼ Instance type Info

4) Select Instance Type

Instance type

t2.micro Free tier eligible
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

Compare instance types

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

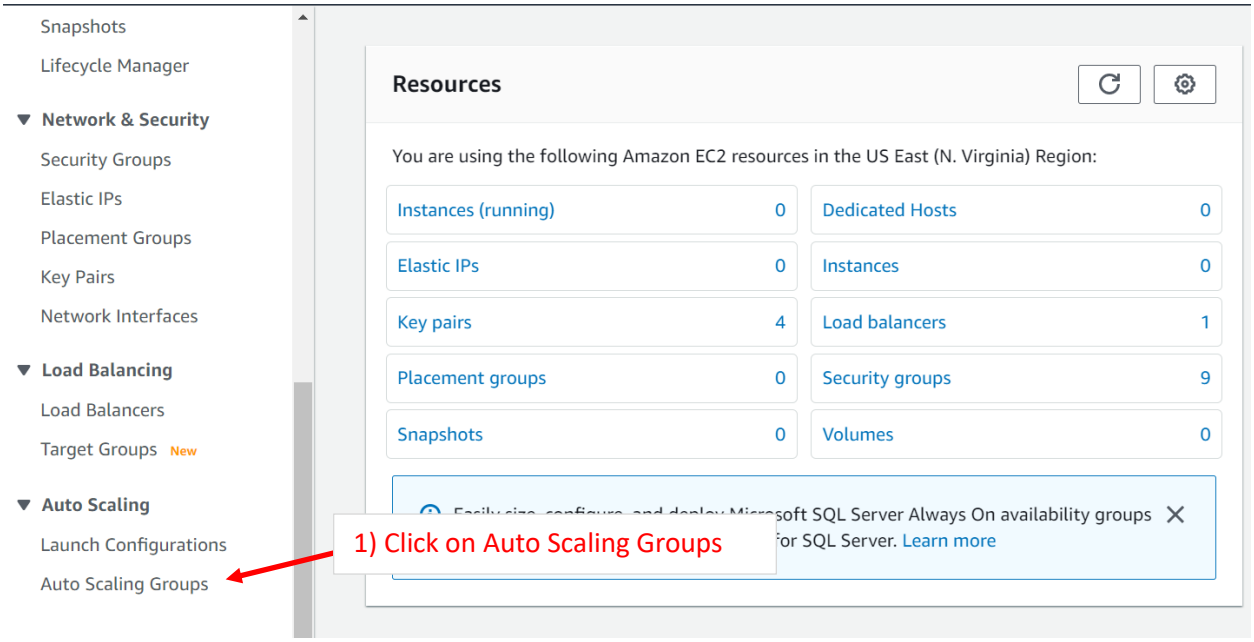
Select security groups

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

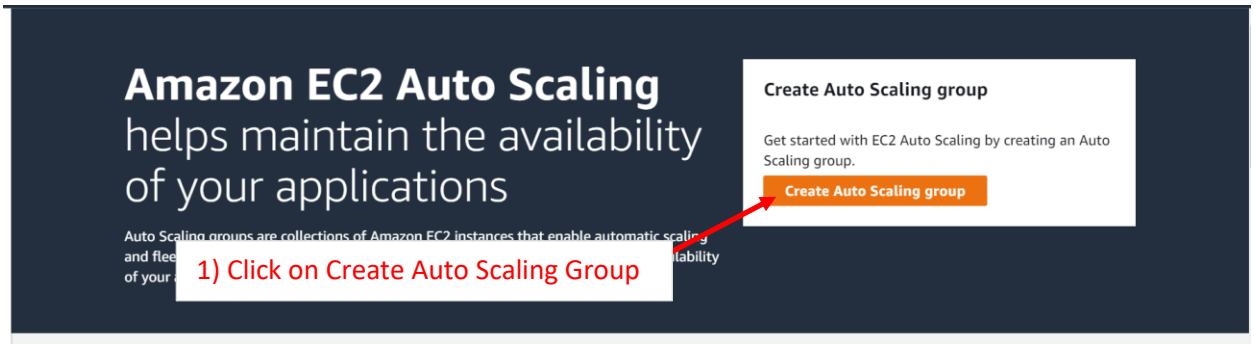
5) Select Server security Group you Created

Create Auto Scaling Group

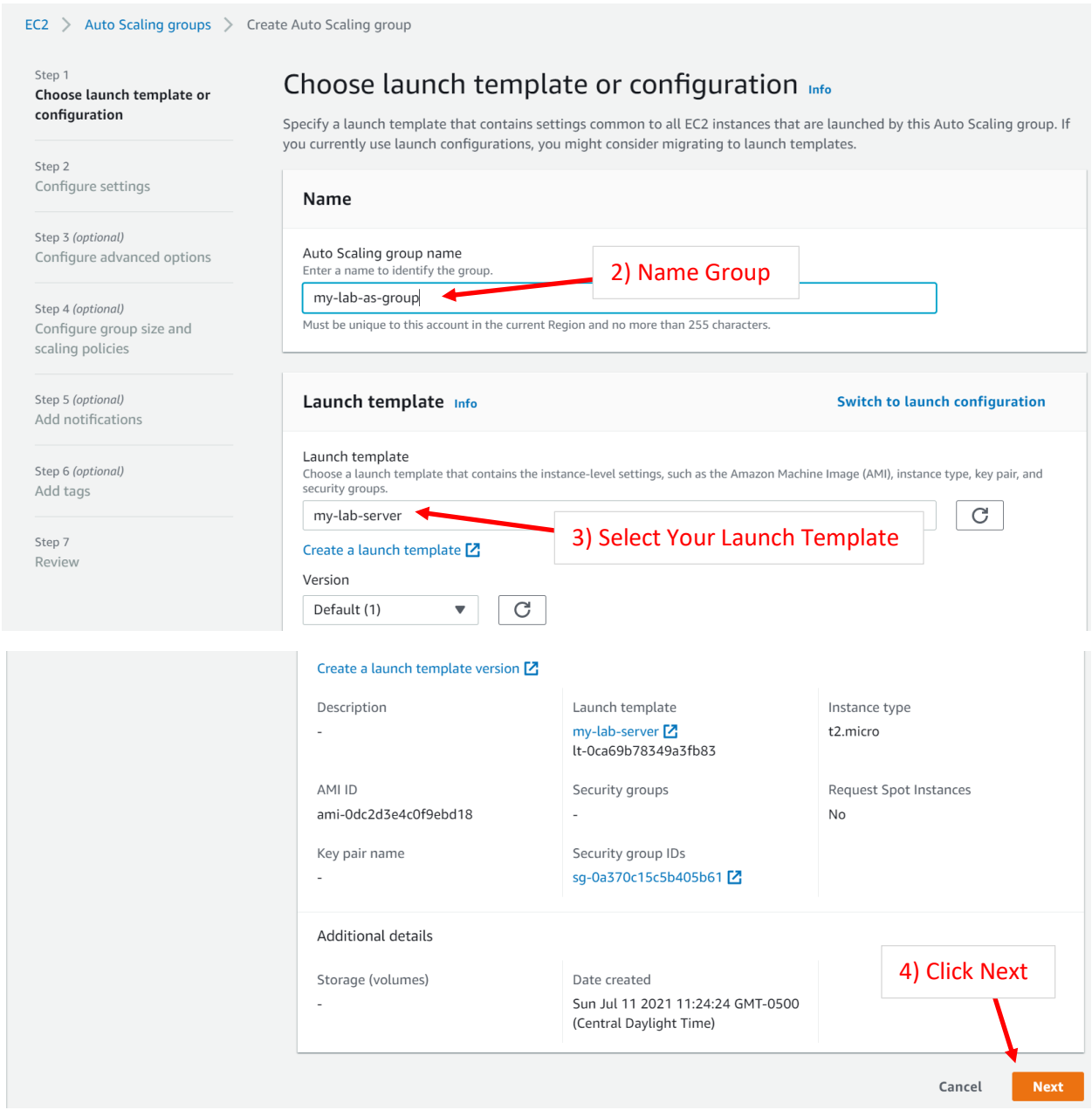
Go to Auto Scaling Display from EC2 Display



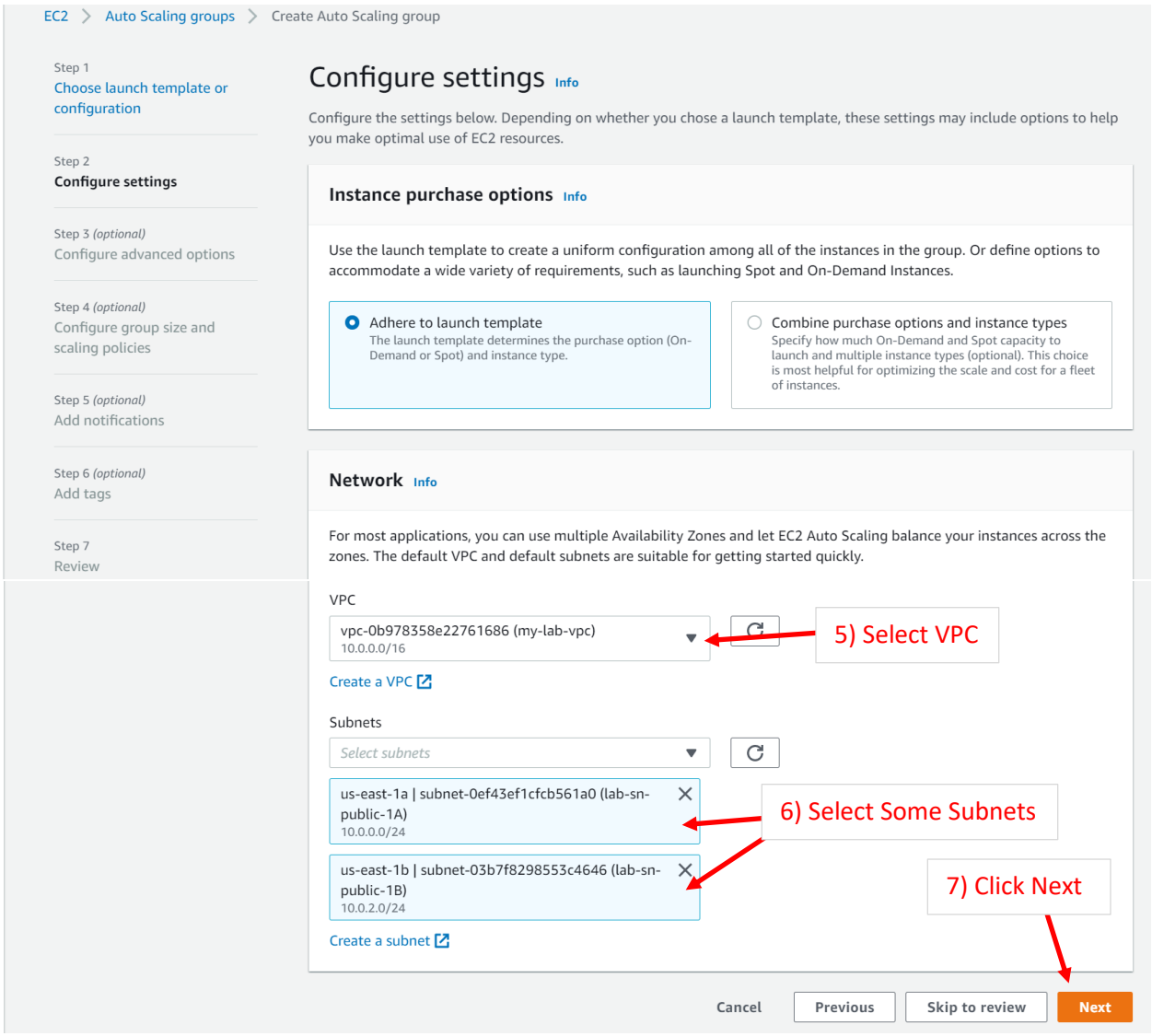
Create Auto Scaling Group



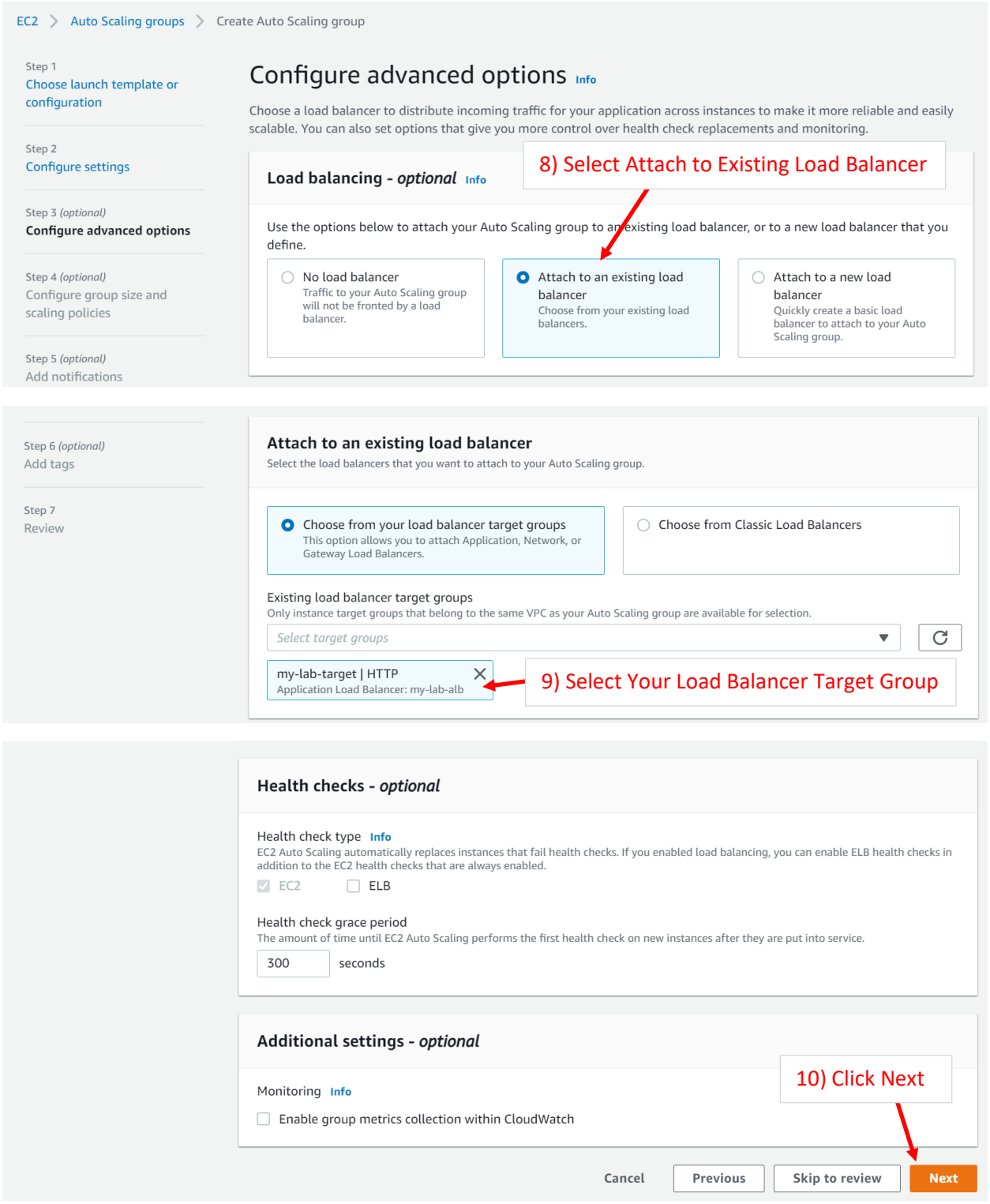
...


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)
Create a launch template version
Description
-
Launch template
my-lab-server
lt-0ca69b78349a3fb83
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)
4) Click Next
Cancel
Next

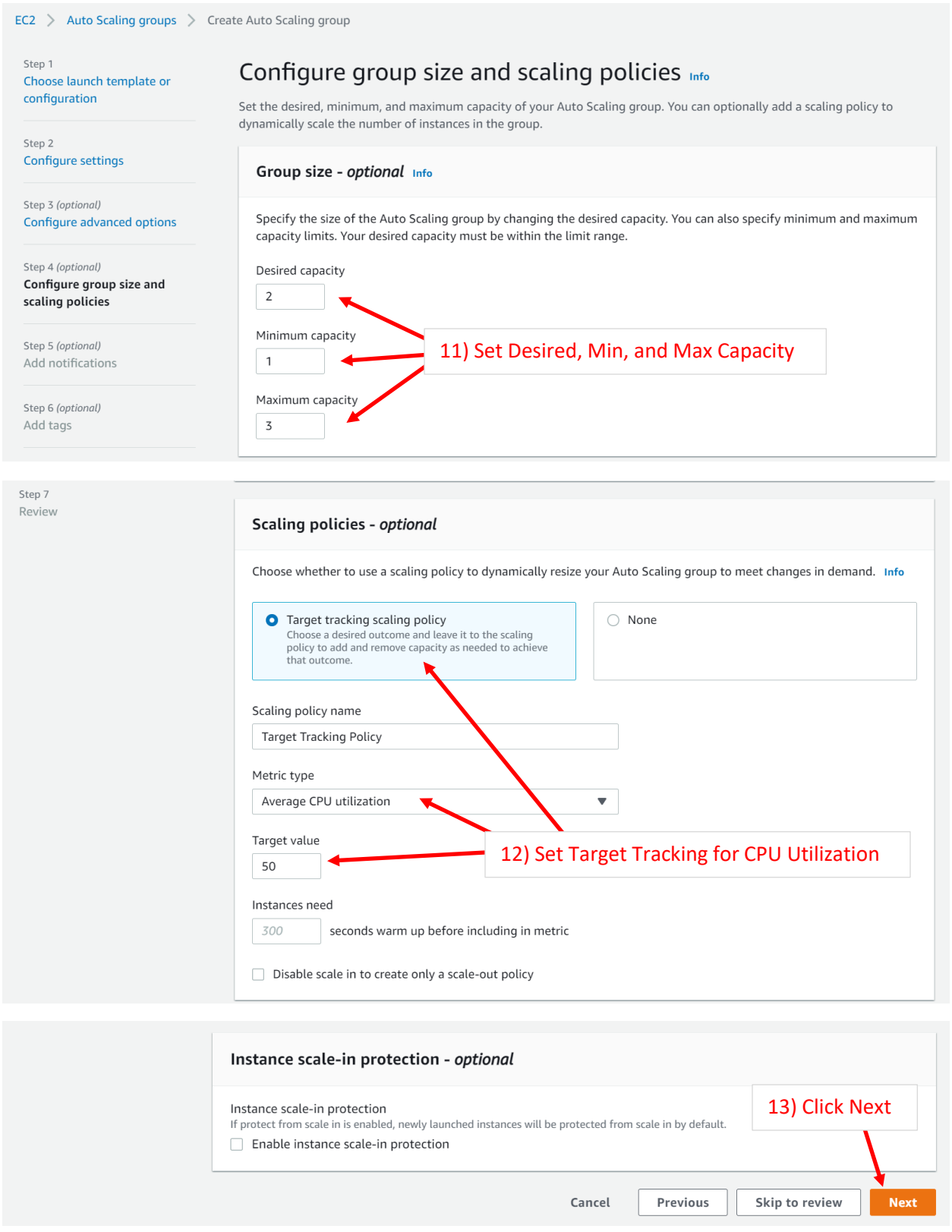
...


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
5) Select VPC
Create a VPC
Subnets
Select subnets
us-east-1a | subnet-0ef43ef1cfcb561a0 (lab-sn-public-1A)
10.0.0.0/24
us-east-1b | subnet-03b7f8298553c4646 (lab-sn-public-1B)
10.0.2.0/24
6) Select Some Subnets
7) Click Next
Create a subnet
Cancel
Previous
Skip to review
Next

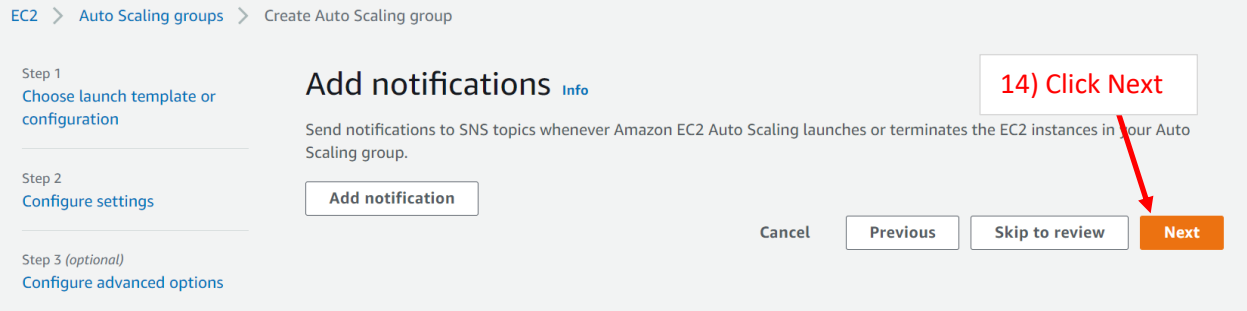
...


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
Configure advanced options Info
Choose a load balancer to distribute incoming traffic for your application across instances to make it more reliable and easily scalable. You can also set options that give you more control over health check replacements and monitoring.
Load balancing - optional Info
8) Select Attach to Existing Load Balancer
Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.
No load balancer
Traffic to your Auto Scaling group will not be fronted by a load balancer.
Attach to an existing load balancer
Choose from your existing load balancers.
Attach to a new load balancer
Quickly create a basic load balancer to attach to your Auto Scaling group.
Step 6 (optional)
Add tags
Step 7
Review
Attach to an existing load balancer
Select the load balancers that you want to attach to your Auto Scaling group.
Choose from your load balancer target groups
This option allows you to attach Application, Network, or Gateway Load Balancers.
Choose from Classic Load Balancers
Existing load balancer target groups
Only instance target groups that belong to the same VPC as your Auto Scaling group are available for selection.
Select target groups
my-lab-target | HTTP
Application Load Balancer: my-lab-alb
9) Select Your Load Balancer Target Group
Health checks - optional
Health check type Info
EC2 Auto Scaling automatically replaces instances that fail health checks. If you enabled load balancing, you can enable ELB health checks in addition to the EC2 health checks that are always enabled.
EC2
ELB
Health check grace period
The amount of time until EC2 Auto Scaling performs the first health check on new instances after they are put into service.
300
seconds
Additional settings - optional
10) Click Next
Monitoring Info
Enable group metrics collection within CloudWatch
Cancel
Previous
Skip to review
Next

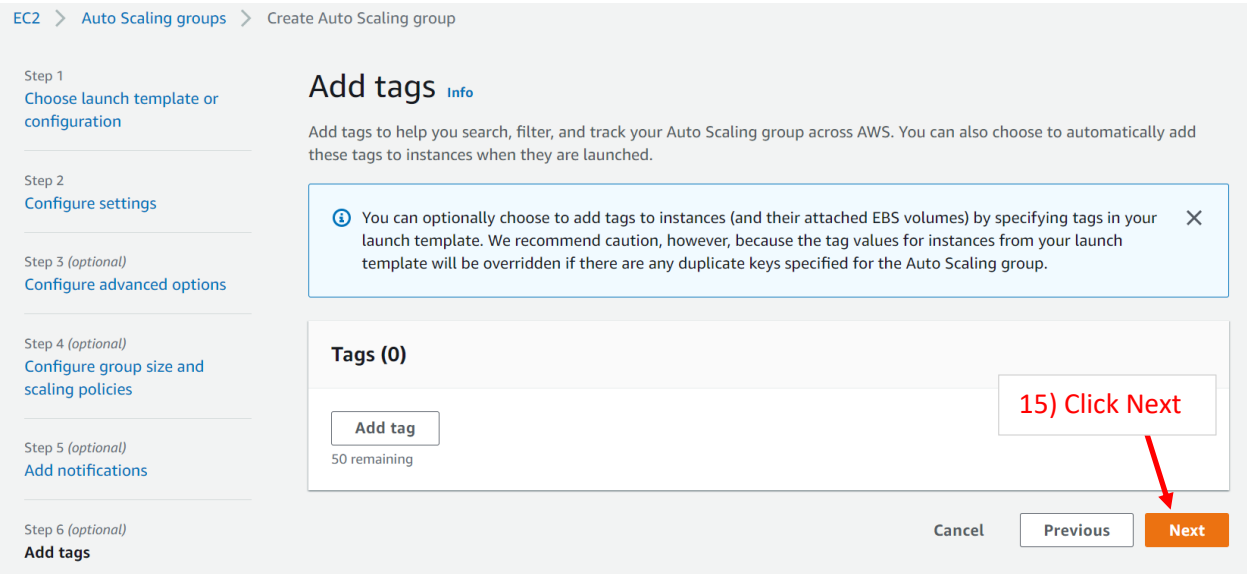
...


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 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

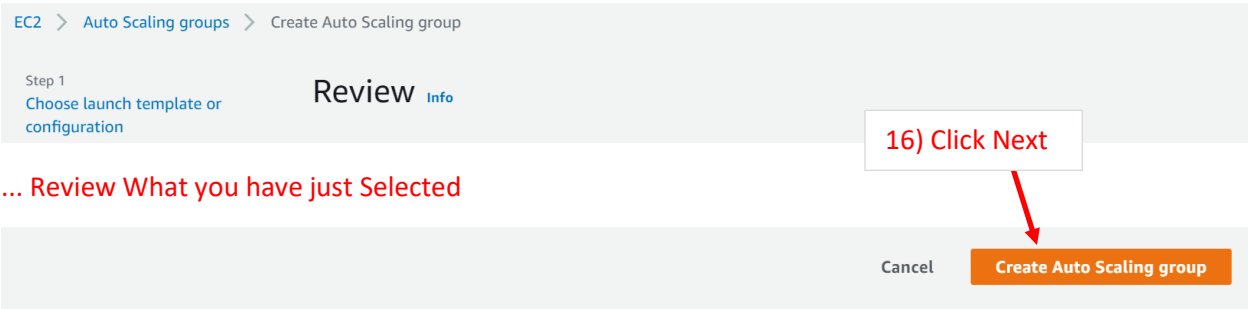
...



...



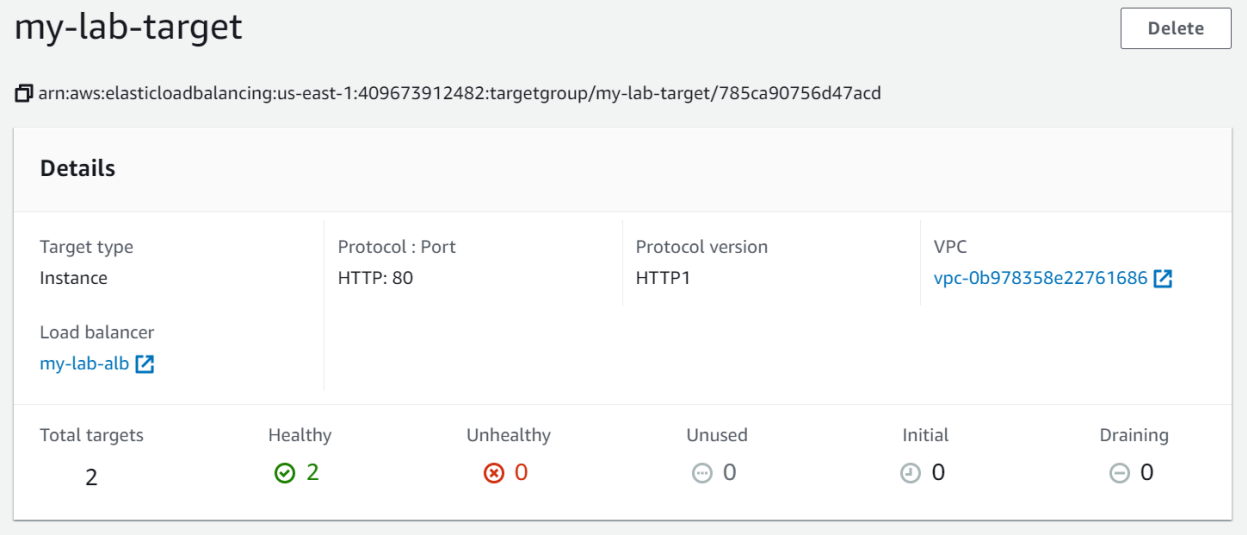
...

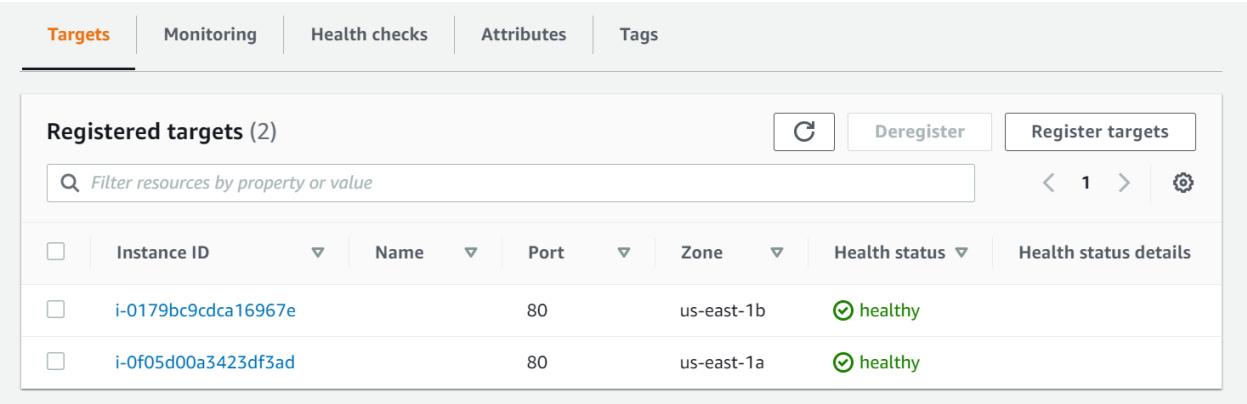


... Review What you have just Selected

Verify and Test the ALB

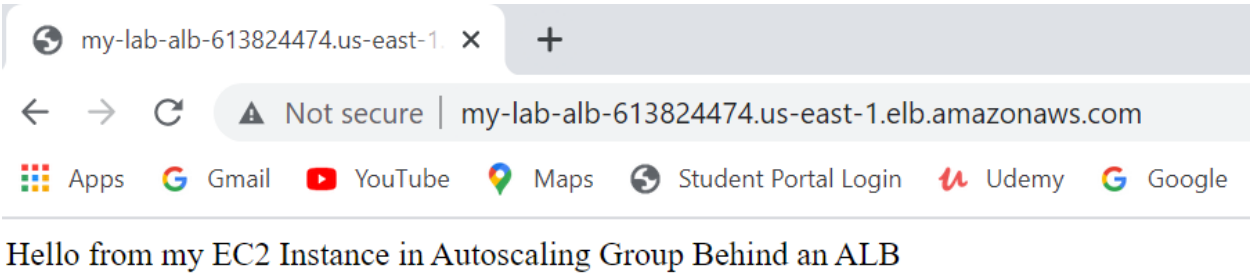
View the Health Check on your the Target Group Details. Both Instances Should be Healthy





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

Test DNS with Web Browser



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.

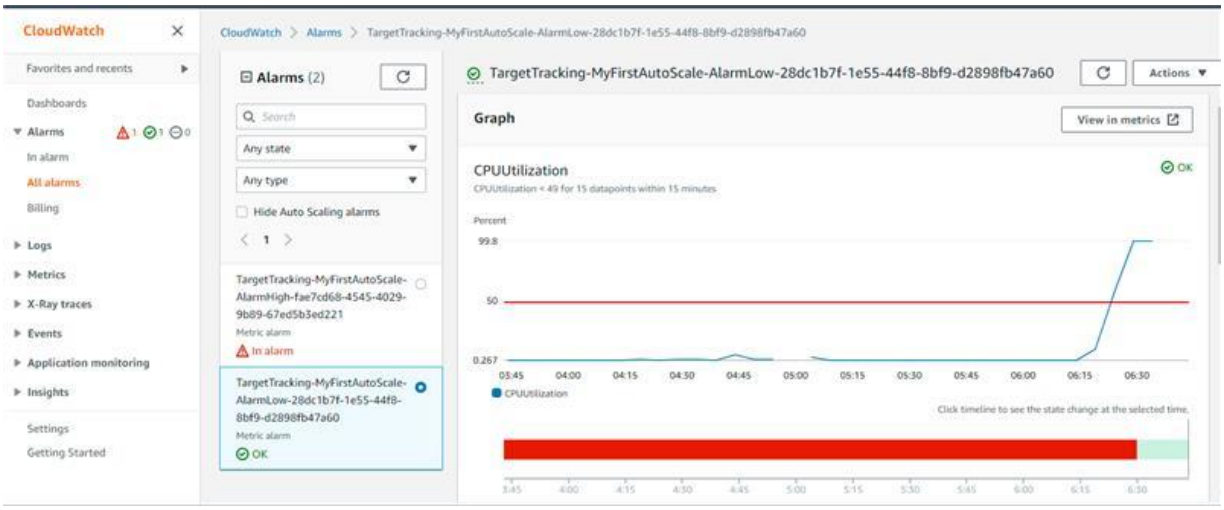
install stress tool using the following commands:

sudo amazon-linux-extras install epel -y

sudo yum install stress -y

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