

AutoScaling & LoadBalancer ASSIGNMENT



Name – Maithely Sharma
College – University of Petroleum and Energy Studies
EmployeeID – 4057

1. Differences between ELB, ALB, and NLB. Where will you use which one?

- **Elastic Load Balancing** - Elastic Load Balancing automatically distributes incoming application traffic across multiple targets, such as Amazon EC2 instances, containers, IP addresses, and Lambda functions. It can handle the varying load of your application traffic in a single Availability Zone or across multiple Availability Zones. Elastic Load Balancing offers three types of load balancers that all feature the high availability, automatic scaling, and robust security necessary to make your applications fault tolerant.
- **Application Load Balancer** - Application Load Balancer is best suited for load balancing of HTTP and HTTPS traffic and provides advanced request routing targeted at the delivery of modern application architectures, including microservices and containers. Operating at the individual request level (Layer 7), Application Load Balancer routes traffic to targets within Amazon Virtual Private Cloud (Amazon VPC) based on the content of the request.
- **Network Load Balancer** - Network Load Balancer is best suited for load balancing of Transmission Control Protocol (TCP), User Datagram Protocol (UDP) and Transport Layer Security (TLS) traffic where extreme performance is required. Operating at the connection level (Layer 4), Network Load Balancer routes traffic to targets within Amazon Virtual Private Cloud (Amazon VPC) and is capable of handling millions of requests per second while maintaining ultra-low latencies. Network Load Balancer is also optimized to handle sudden and volatile traffic patterns.

2.Differences between step scaling and target scaling.

With **step** scaling and **simple** scaling, you choose scaling metrics and threshold values for the CloudWatch alarms that trigger the scaling process. You also define how your Auto Scaling group should be scaled when a threshold is in breach for a specified number of evaluation periods.

AWS recommends that you use a **target tracking scaling policy** to scale on a metric like **average CPU utilization** or the **RequestCountPerTarget** metric from the Application Load Balancer. Metrics that decrease when capacity increases and increase when capacity decreases can be used to proportionally scale out or in the number of instances using target tracking. This helps ensure that Amazon EC2 Auto Scaling follows the demand curve for your applications closely.

With **target tracking scaling policies**, you select a scaling metric and set a target value. Amazon EC2 Auto Scaling creates and manages the CloudWatch alarms that trigger the scaling policy and calculates the scaling adjustment based on the metric and the target value. The scaling policy adds or removes capacity as required to keep the metric at, or close to, the specified target value. In addition to keeping the metric close to the target value, a target tracking scaling policy also adjusts to the changes in the metric due to a changing load pattern.

3.Differences between Launch configuration and launch template.

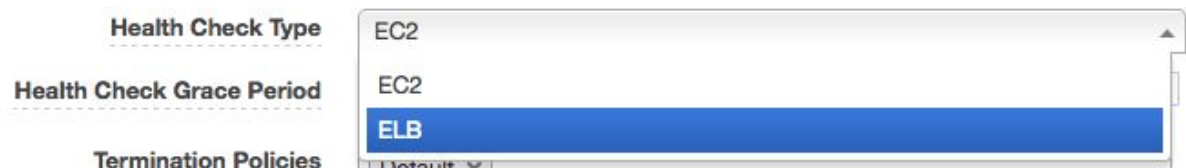
Launch template is similar to launch configuration which usually Auto Scaling group uses to launch EC2 instances. However, defining a launch template instead of a launch configuration allows you to have multiple versions of a template.

AWS recommend that we should use launch templates instead of launch configurations to ensure that we can leverage the latest features of Amazon EC2, such as T2 Unlimited instances.

With launch templates, you can also provision capacity across multiple instance types using both On-Demand Instances and Spot Instances to achieve the desired scale, performance, and cost.

4. Differences between EC2 healthcheck and load balancer health check

Once your ELB has been associated with the auto scaling group, you can then instruct it to be used for the Health Check Type:



EC2 health check watches for instance availability from hypervisor and networking point of view. For example, in case of a hardware problem, the check will fail.

ELB health check verifies that a specified TCP port on an instance is accepting connections OR a specified web page returns 2xx code. Thus ELB health checks are a little bit smarter and verify that actual app works instead of verifying that just an instance works.

5. Create 2 auto-scaling groups with

launch configuration and
launch template

Create a launch configuration

Launch configuration creation status

✓ Successfully created launch configuration: **maithely-conf**
[View creation log](#)

View

[View your launch configurations](#)
[View your Auto Scaling groups](#)

► Here are some helpful resources to get you started

[Create an Auto Scaling group using this launch configuration](#) [Close](#)

Using the above launch configuration, create an autoscaling group

[1. Configure Auto Scaling group details](#) [2. Configure scaling policies](#) [3. Configure Notifications](#) [4. Configure Tags](#) [5. Review](#)

Cancel and Exit

Create Auto Scaling Group

Group name ⓘ autoscaling-maithely

Launch Configuration ⓘ maithely-conf

Group size ⓘ Start with 1 instances

Network ⓘ vpc-d38d68b7 (172.31.0.0/16) | default (default) [Create new VPC](#)

Subnet ⓘ subnet-06680a5b651f104dc(172.31.0.0/16) | default | [Create new subnet](#)

⚠ No public IP addresses will be assigned

None of the instances in this Auto Scaling group will be assigned a public IP address because you have not chosen to launch in your default VPC and subnet.

Now create a launch template

Create launch template

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

Launch template name and description

Launch template name - *required*

maithely-template

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

Template version description

template created

Max 255 chars

Auto scaling guidance [Info](#)

Select this if you intend to use this template with auto scaling

☒ Provide guidance to help me set up a template that I can use with auto scaling

► Template tags

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

Ubuntu Server 18.04 LTS (HVM), SSD Volume Type

ami-07ebfd5b3428b6f4d

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

Instance type [Info](#)

Instance type

t2.micro

Family: General purpose 1 vCPU 1 GiB Memory

On-Demand Linux pricing: 0.0116 USD per Hour

On-Demand Windows pricing: 0.0162 USD per Hour

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 [Info](#)

Select security groups

AutoScaling-Security-Group-maithely sg-0f97d62f616939b00 X

VPC: vpc-d38d68b7

Storage (volumes) [Info](#)

► Volume 1 (AMI Root) (8 GiB, EBS, General purpose SSD (gp2))

AMI Volumes are not included in the template unless modified

Add new volume

After creating the launch template , now attach it to the auto scaling group

The screenshot shows the 'Edit details' dialog for an AWS Auto Scaling group named 'autoscaling-maithely'. The dialog is a light gray box with a close button (X) in the top right corner. It contains several sections for configuring the autoscaling group:

- Launch Instances Using:** Two radio buttons are present. 'Launch Template' is selected, and 'Launch Configuration' is unselected.
- Launch Template:** A dropdown menu showing 'maithely-template'.
- Launch Template Version:** A dropdown menu showing 'Latest'.
- Launch Template Description:** A text field containing 'template created'.
- Fleet Composition:** Two radio buttons are present. 'Adhere to the launch template' is selected. Below it, a text description reads: 'The launch template determines the instance type and purchase option (On-Demand or Spot)'. The second radio button, 'Combine purchase options and instances', is unselected. Below it, a text description reads: 'Choose a mix of On-Demand Instances and Spot Instances and multiple instance types. Spot Instances are automatically launched at the lowest price available.'
- Desired Capacity:** A text input field containing the number '1'.
- Min:** A text input field containing the number '1'.
- Max:** A text input field containing the number '1'.
- Availability Zone(s):** A text input field containing 'us-east-1c' with a small 'x' icon to its right.

At the bottom right of the dialog, there are two buttons: 'Cancel' (light gray) and 'Save' (blue).

6. Setup autoscaling Wordpress application with the Application load balancer. Auto-scaling should be triggered based on CPU usage of EC2 instances. Create an instance and install install wordpress with nginx and mysql: <https://www.journaldev.com/25670/install-wordpress-nginx-ubuntu>

Launch Instance Connect Actions

search : maithely Add filter 1 to 2 of 2

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status
maithely-ec2	i-0a5b4f4e3b765207c	t2.micro	us-east-1c	running	2/2 checks ...	None
	i-0a95cbe2ff87da7da	t2.micro	us-east-1c	running	2/2 checks ...	None

Instance: i-0a5b4f4e3b765207c (maithely-ec2) Public DNS: ec2-3-86-84-146.compute-1.amazonaws.com

Description Status Checks Monitoring Tags

Instance ID i-0a5b4f4e3b765207c Public DNS (IPv4) ec2-3-86-84-146.compute-1.amazonaws.com

Instance state running IPv4 Public IP 3.86.84.146

Instance type t2.micro IPv6 IPs -

Finding You may not have permission to access AWS Compute Optimizer. Elastic IPs

Private DNS ip-172-31-221-243.ec2.internal Availability zone us-east-1c

Private IPs 172.31.221.243 Security groups AutoScaling-Security-Group-maithely. view inbound rules. view outbound rules

← → ↻ ⓘ Not secure | 3.86.84.146/wp-admin/ myblog1 0 + New

Dashboard

Home Updates

Posts Media Pages Comments Appearance Plugins Users Tools Settings Collapse menu

Dashboard

Welcome to WordPress!
We've assembled some links to get you started:

Get Started

[Customize Your Site](#)

or, [change your theme completely.](#)

Next Steps

- [Write your first blog post](#)
- [Add an About page](#)
- [Set up your homepage](#)
- [View your site](#)

More Actions

- [Manage widgets](#)
- [Manage menus](#)
- [Turn comments on or off](#)
- [Learn more about getting started](#)

At a Glance

1 Post 1 Page

1 Comment

WordPress 5.3.2 running [Twenty Twenty](#) theme.

Quick Draft

Title

Content

What's on your mind?

Activity

Now create a vpc and attach IGW to it

Create VPC Actions

VPC ID : vpc-0337162a6a8303c7d Add filter 1 to 1 of 1

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP options set	Main Route table
maithely-vpc	vpc-0337162a6a8303c7d	available	10.0.0.0/16	-	dopt-519d6f34	rtb-06f83cae230538e

Create internet gateway Actions

Filter by tags and attributes or search by keyword

<input type="checkbox"/>	Name	ID	State	VPC	Owner
<input checked="" type="checkbox"/>	maithely-igw	igw-08071c47af30...	attached	vpc-0337162a6a8...	187632318301
<input type="checkbox"/>		igw-0852d164026...	attached	vpc-0b061c711cd...	187632318301

Now create two subnets in different AZs

Create subnet Actions

search : maithely Add filter

<input type="checkbox"/>	Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4
<input checked="" type="checkbox"/>	maithely-subnet1	subnet-04a109cc7cd6b5616	available	vpc-0337162a6a8303c7d ...	10.0.0.0/24	250
<input type="checkbox"/>	maithely-subnet2	subnet-058b99f2a73cfd1a1	available	vpc-0337162a6a8303c7d ...	10.0.1.0/24	251
<input checked="" type="checkbox"/>	maithely-subnet3	subnet-0a2fce22263b58524	available	vpc-0337162a6a8303c7d ...	10.0.2.0/24	251

Attach the subnets in the route table

Create route table Actions

search : maithely Add filter

<input checked="" type="checkbox"/>	Name	Route Table ID	Explicit subnet association	Edge associations	Main
<input checked="" type="checkbox"/>		rtb-06f83cae230538e06	3 subnets	-	Yes

Route Table: rtb-06f83cae230538e06

Summary Routes Subnet Associations Edge Associations Route Propagation Tags

Edit subnet associations

Subnet ID	IPv4 CIDR	IPv6 CIDR
subnet-04a109cc7cd6b56...	10.0.0.0/24	-
subnet-0a2fce22263b585...	10.0.2.0/24	-
subnet-058b99f2a73cfd1a...	10.0.1.0/24	-

The following subnets have not been explicitly associated with any route tables and are therefore associated with the main route table:

Now create an AMI of the instance created

Create Image

Instance ID

i-0a5b4f4e3b765207c

Image name

wordpress-maithely

Image description

wordpress

No reboot

☐

Instance Volumes

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encrypted
Root	/dev/sda1	snap-0e078112eedec9db	8	General Purpose SSD (gp2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

Add New Volume

Total size of EBS Volumes: 8 GiB
When you create an EBS image, an EBS snapshot will also be created for each of the above volumes.

Cancel

Create Image

Now create an configuration template and attach AMI to it

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*

maithely-wordpress-template

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

Template version description

template

Max 255 chars

Auto scaling guidance [Info](#)

Select this if you intend to use this template with auto scaling

☐ Provide guidance to help me set up a template that I can use with auto scaling

Template tags

Now create auto scaling group

Filter: <input type="text" value="Filter Auto Scaling groups..."/>		1 to 3 of 3 Auto Scaling Groups						
<input type="checkbox"/>	Name	Launch Configuration /	Instances	Desired	Min	Max	Availability Zones	Default Cool
<input checked="" type="checkbox"/>	autoscaling-maithely-wordpress	maithely-wordpress-te...	0	1	1	1	us-east-1c	300
Launch Template		maithely-wordpress-template	1	2	1	2	us-east-1c	300
Launch Template Version		1	Subnet(s)		subnet-06680a5b65			
Launch Template Description		template	Classic Load Balancers					
Instance Types		-	Target Groups					
Spot Allocation Strategy			Health Check Type		EC2			
Optional On-Demand Base		0	Health Check Grace Period		300			
On-Demand Percentage		0%	Instance Protection					
Desired Capacity		1	Termination Policies		Default			
Min		1	Suspended Processes					
		-	Max Instance Lifetime					

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

Add listener

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 must specify subnets from at least two Availability Zones to increase the availability of your load balancer.

VPC

Availability Zones
☒ us-east-1b
☒ us-east-1c

IPv4 address Assigned by AWS

Now you have to edit the ASG and add target group

Edit details - autoscaling-maithely-wordpress

max

Availability Zone(s)

us-east-1c

Subnet(s)

subnet-04a109cc7cd6b5616(10.0.0.0/24) | maithely-subnet1 | us-east-1c
subnet-058b99f2a73cfd1a1(10.0.1.0/24) | maithely-subnet2 | us-east-1c

Classic Load Balancers

Target Groups

load-maithely

Health Check Type

EC2

Health Check Grace Period

300

Instance Protection

Termination Policies

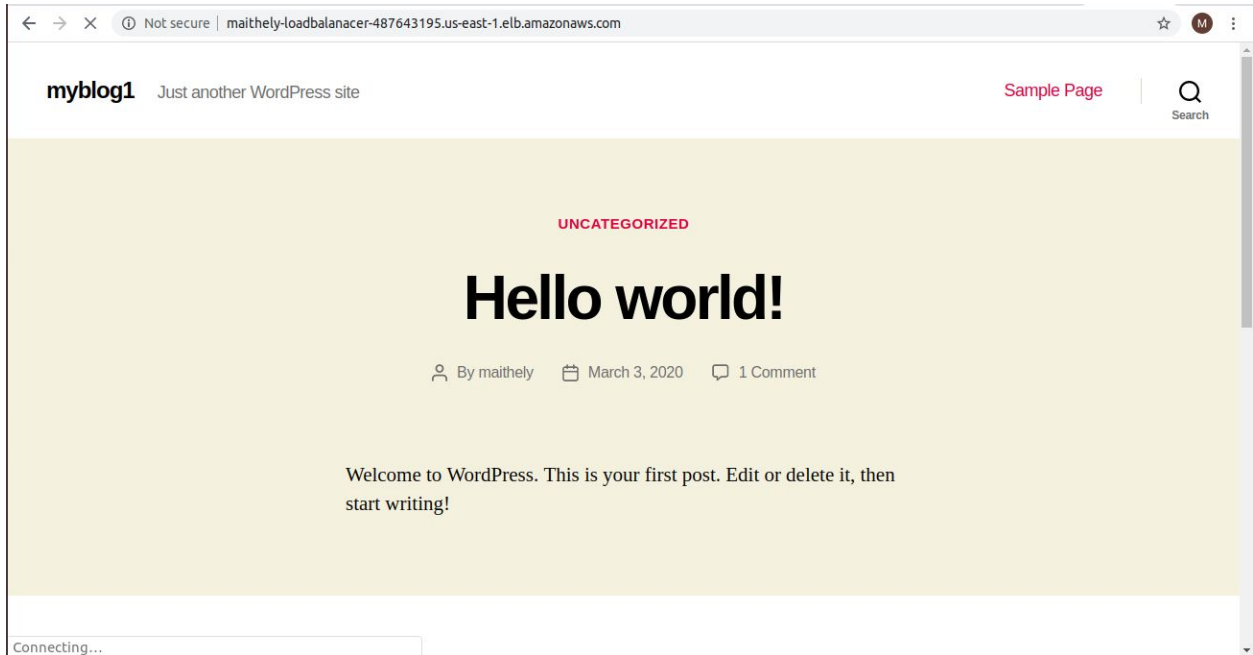
Default

Suspended Processes

Cancel

Save

Browse the DNS of load balancer



7. Create another Wordpress website and use the ALB created above to send traffic to this website based on the hostname

Create 2 instances

A screenshot of the AWS Management Console showing a list of EC2 instances. The table has columns for Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Status, and Public D. There are three instances listed: 'maithely-ec2-1' (running), 'maithely-ec2-2' (initializing), and an unnamed instance (running). Below the table, there are tabs for 'Description', 'Status Checks', 'Monitoring', and 'Tags'. The 'Description' tab is selected, showing the instance IDs 'i-0d2092049786646df' and 'i-0414b721dcbf9d63a'.

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public D
maithely-ec2-1	i-0414b721dcbf9d63a	t2.micro	us-east-1c	running	2/2 checks ...	None	
maithely-ec2-2	i-0d2092049786646df	t2.micro	us-east-1c	running	Initializing	None	
	i-0d6921d7e0c9df1b0	t2.micro	us-east-1c	running	2/2 checks ...	None	

Instances: i-0d2092049786646df (maithely-ec2-2), i-0414b721dcbf9d63a (maithely-ec2-1)

Description Status Checks Monitoring Tags

i-0d2092049786646df:
i-0414b721dcbf9d63a:

Now ssh in both the instances and install nginx


```
maithely@maithely:~/Downloads$ ssh -i "maithely.pem" ubuntu@54.205.14.30

The authenticity of host '54.205.14.30 (54.205.14.30)' can't be established.
ECDSA key fingerprint is SHA256:MLHRuMavfUM3/ZAlaoMRfV82IaFmUIX400toxQkKoU0.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '54.205.14.30' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-1057-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Wed Mar  4 10:04:55 UTC 2020

System load:  0.04               Processes:            89
Usage of /:   13.6% of 7.69GB    Users logged in:     0
Memory usage: 15%               IP address for eth0: 10.0.0.35
Swap usage:   0%

0 packages can be updated.
0 updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
```

```
ubuntu@ip-10-0-0-35:~$ sudo apt install nginx
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  fontconfig-config fonts-dejavu-core libfontconfig1 libfontconfig1-dev
  libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter
  nginx-core
Suggested packages:
  libgd-tools fcgiwrap nginx-doc ssl-cert
```

And also create a directory app1 and a file index.html

```
ubuntu@ip-10-0-0-35:~$ cd /var/www/html/app1/
ubuntu@ip-10-0-0-35:/var/www/html/app1$ sudo vim index.html
ubuntu@ip-10-0-0-35:/var/www/html/app1$ ls
index.html
```

Similarly do the same for the second instance as well.

Create 2 target groups

Create target group

Your load balancer routes requests to the targets in a target group using the target group settings that you specify, and performs health checks on the targets using the health check settings that you specify.

Target group name

maithelytg1

Target type

Instance

IP

Lambda function

Protocol

HTTP

Port

80

VPC

vpc-d38d68b7 (172.31.0.0/16) | default (My I

Health check settings

Protocol

HTTP

Path

/app1

Create target group

Actions

Filter by tags and attributes or search by keyword

1 to 3 of 3

	Name	Port	Protocol	Target type	Load Balanc	VPC ID	Monitoring
<input type="checkbox"/>	load-maithely	80	HTTP	instance	maithely-lo...	vpc-0337162a6a8303c7d	
<input checked="" type="checkbox"/>	maithelytg2	80	HTTP	instance		vpc-d38d68b7	
<input checked="" type="checkbox"/>	maithely1tg	80	HTTP	instance	maithely-lo...	vpc-0337162a6a8303c7d	

Target groups: maithely1tg, maithelytg2

After creating the target groups edit to add instance

Registered targets

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

[Remove](#)

<input type="checkbox"/>	Instance	Name	Port	State	Security groups	Zone
<input checked="" type="checkbox"/>	i-0d209204978664...	maithely-ec2-2	80	running	maithely-wordpress-sg	us-east-1c

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

[X](#)

<input type="checkbox"/>	Instance	Name	State	Security	Zone	Subnet ID	Subnet CIDR
<input type="checkbox"/>	i-0d6921d7e...		running	maithely-wo...	us-east-1c	subnet-04a109cc7cd6b5616	10.0.0.0/24
<input type="checkbox"/>	i-0414b721d...	maithely-ec2-1	running	maithely-wo...	us-east-1c	subnet-04a109cc7cd6b5616	10.0.0.0/24
<input checked="" type="checkbox"/>	i-0d2092049...	maithely-ec2-2	running	maithely-wo...	us-east-1c	subnet-04a109cc7cd6b5616	10.0.0.0/24

In the load balancer edit the rules in the listener

[Create Load Balancer](#) [Actions](#)

<input checked="" type="checkbox"/>	Name	DNS name	State
<input checked="" type="checkbox"/>	maithely-loadbalancer	maithely-loadbalancer-487643195.us-east-1.elb.amazonaws.com	active

Load balancer: **maithely-loadbalancer**

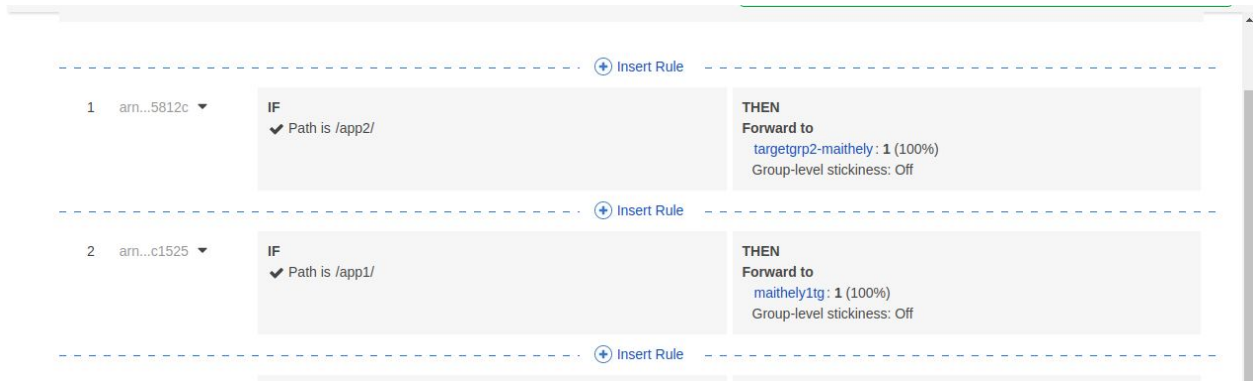
[Description](#) [Listeners](#) [Monitoring](#) [Integrated services](#) [Tags](#)

A listener checks for connection requests using its configured protocol and port, and the load balancer uses the listener rules to route requests. You can add, remove, or update listeners and listener rules.

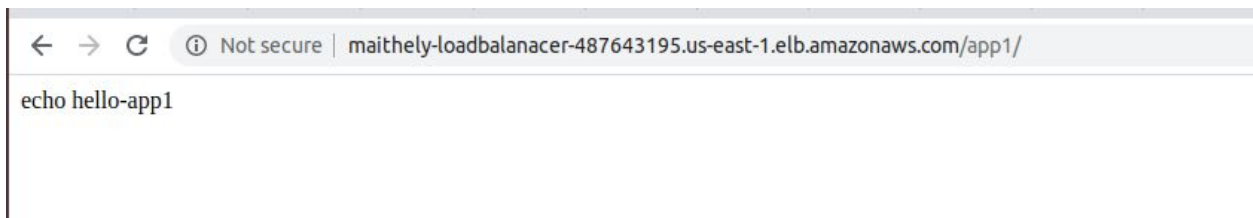
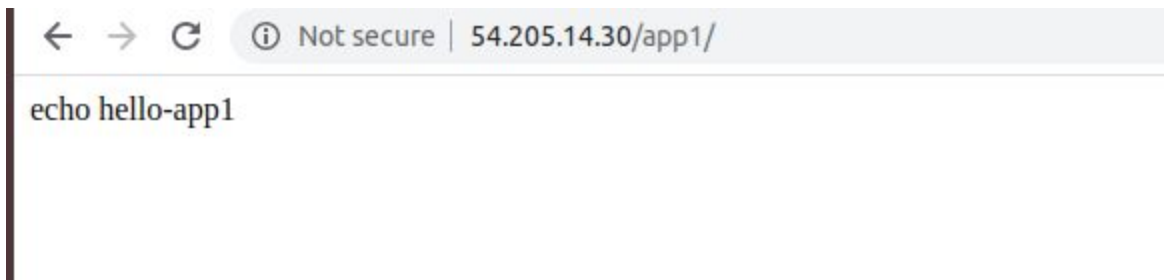
[Add listener](#) [Edit](#) [Delete](#)

<input type="checkbox"/>	Listener ID	Security policy	SSL Certificate	Rules
<input type="checkbox"/>	HTTP : 80 arn:...3f31c6fa598c7216	N/A	N/A	Default: forwarding to load-maithely View/edit rules

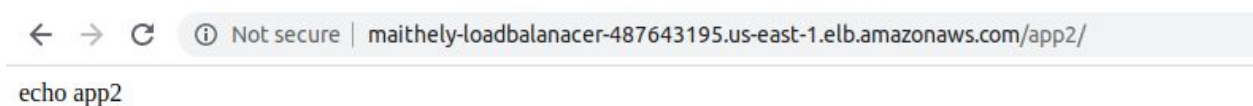
Where you add the actions



Now browse at ip at app1



Now browse at ip at app2



8. Use NLB that replaces the ALB in the above setup.
Create a network load balancer

Step 1: Configure 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
TCP	80

Add listener

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 may also add one Elastic IP per Availability Zone if you wish to have specific addresses for your load balancer.

Create and manage Elastic IPs in the VPC console

VPC *i* vpc-0337162a6a8303c7d (10.0.0.0/16) | maithely-vpc

Availability Zones ☒ us-east-1b subnet-0a2fce22263b58524 (maithely-subnet3)

Cancel Next: Configure Security Settings

Now create a target group

Create target group Actions

Filter by tags and attributes or search by keyword							
<input type="checkbox"/>	Name	Port	Protocol	Target type	Load Balancer	VPC ID	
<input checked="" type="checkbox"/>	network-lb	80	TCP	instance	networklb-...	vpc-0337162a6a8303c7d	
<input type="checkbox"/>	networktg-maithely1	80	HTTP	instance		vpc-d38d68b7	

Now add target group to the asg

Edit details - autoscaling-maithely-wordpress



.....

Max 1

Availability Zone(s) us-east-1c

Subnet(s)

subnet-04a109cc7cd6b5616(10.0.0.0/24) | maithely-subnet1 | us-east-1c

subnet-058b99f2a73cfd1a1(10.0.1.0/24) | maithely-subnet2 | us-east-1c

Classic Load Balancers

Target Groups network-lb |

Health Check Type EC2

Health Check Grace Period 300

Instance Protection

Termination Policies Default

Cancel

Save

Filter: <input type="text" value="Filter Auto Scaling groups..."/>									
1 to 3 of 3 Auto Scaling Groups									
<input type="checkbox"/>	Name	Launch Configuration /	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health Ch
<input type="checkbox"/>	eks-68b85232-...	eks-68b85232-8a58-38...	2	2	2	2	us-east-1a, us-east-1b, us-e...	300	15
<input checked="" type="checkbox"/>	autoscaling-m...	maithely-template-word...	1	1	1	1	us-east-1c	300	300
<input type="checkbox"/>	revant	revant	0	0	0	0	us-east-1c	300	300

Auto Scaling Group: autoscaling-maithely-wordpress									
Details	Activity History	Scaling Policies	Instances	Monitoring	Notifications	Tags	Scheduled Actions	Lifecycle Hooks	
Actions									

Filter: Any Health Status Any Lifecycle State <input type="text" value="Filter instances..."/>									
1 to 1 of 1 Instances									
<input type="checkbox"/>	Instance ID	Lifecycle	Launch Configuration / Template	Availability Zone	Health Status	Protected from			
<input type="checkbox"/>	i-0d6921d7e0c9df1b0	InService	maithely-template-wordpress	us-east-1c	Healthy				

If we are hitting at the ip of the instance we get the app1

←

→

↻

ⓘ

Not secure | 54.205.14.30/app1/

echo hello-app1

But not with DNS because path based at NLB cannot happen at layer 4 -transport layer tcp/ip

←

→

↻

ⓘ

Not secure | networklb-maithely-de004b468f64b399.elb.us-east-1.amazonaws.com/app1

404 Not Found

nginx/1.14.0 (Ubuntu)

9.Take an instance out of the ASG.

Create Auto Scaling group Actions

Filter: Filter Auto Scaling groups... 1 to 3 of 3 Auto Scaling Groups

Name	Launch Configuration / Template	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health Check
eks-68b85232-...	eks-68b85232-8a58-38...	2	2	2	2	us-east-1a, us-east-1b, us-e...	300	15
autoscaling-m...	maithely-template-word...	1	1	1	1	us-east-1c	300	300
revant	revant	0	0	0	0	us-east-1c	300	300

Auto Scaling Group: autoscaling-maithely-wordpress

Details Activity History Scaling Policies Instances Monitoring Notifications Tags Scheduled Actions Lifecycle Hooks

Actions

- Detach
- Set to Standby
- Set to InService
- Instance Protection

1 to 1 of 1 Instances

Instance ID	Lifecycle State	Launch Configuration / Template	Availability Zone	Health Status	Protected from
i-0d6921d7e0c9df1b0	InService	maithely-template-wordpress	us-east-1c	Healthy	

10. Put scale-in protection on an instance in the ASG.

Auto Scaling Group: autoscaling-maithely-wordpress

Details Activity History Scaling Policies Instances Monitoring Notif

Actions

- Detach
- Set to Standby
- Set to InService
- Instance Protection

Set Scale In Protection

Remove Scale In Protection

Create Auto Scaling group Actions

Filter: Filter Auto Scaling groups... 1 to 3 of 3 Auto Scaling Groups

Name	Launch Configuration /	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health
eks-68b85232-...	eks-68b85232-8a58-38...	2	2	2	2	us-east-1a, us-east-1b, us-e...	300	15
autoscaling-m...	maithely-template-word...	1	1	1	1	us-east-1c	300	300
revant	revant	0	0	0	0	us-east-1c	300	300

Auto Scaling Group: autoscaling-maithely-wordpress

Details Activity History Scaling Policies Instances Monitoring Notifications Tags Scheduled Actions Lifecycle Hooks

Actions

Filter: Any Health Status Any Lifecycle State Filter instances... 1 to 1 of 1 Instances

Instance ID	Lifecycle	Launch Configuration / Template	Availability Zone	Health Status	Protected from
i-0dc0a7f750c7d89b1	InService	maithely-template-wordpress	us-east-1c	Healthy	Scale In

11.Put Schedules in ASG to:

Remove all instances of the ASG at 8 PM

Create Scheduled Action

Name scheduler1

Auto Scaling Group autoscaling-maithely-wordpress

Provide at least one of Min, Max and Desired Capacity

Min 0

Max 0

Desired Capacity 0

Recurrence Cron 0 20 * * * Example: 0 23 * * MON-FRI

Start Time 00 : 00 UTC Specify the start time in UTC
The first time this scheduled action will run

End Time Set End Time

Cancel Create

Launch a minimum of 2 instances at 10 AM

Create Scheduled Action



Name

schduler2

Auto Scaling Group

autoscaling-maithely-wordpress

Provide at least one of Min, Max and Desired Capacity

Min

2

Max

5

Desired Capacity

2

Recurrence

Cron

0 10 ***

Example: 0 23 * * MON-FRI

Start Time

00 : 00

UTC

Specify the start time in UTC

The first time this scheduled action will run

End Time

[Set End Time](#)

Cancel

Create