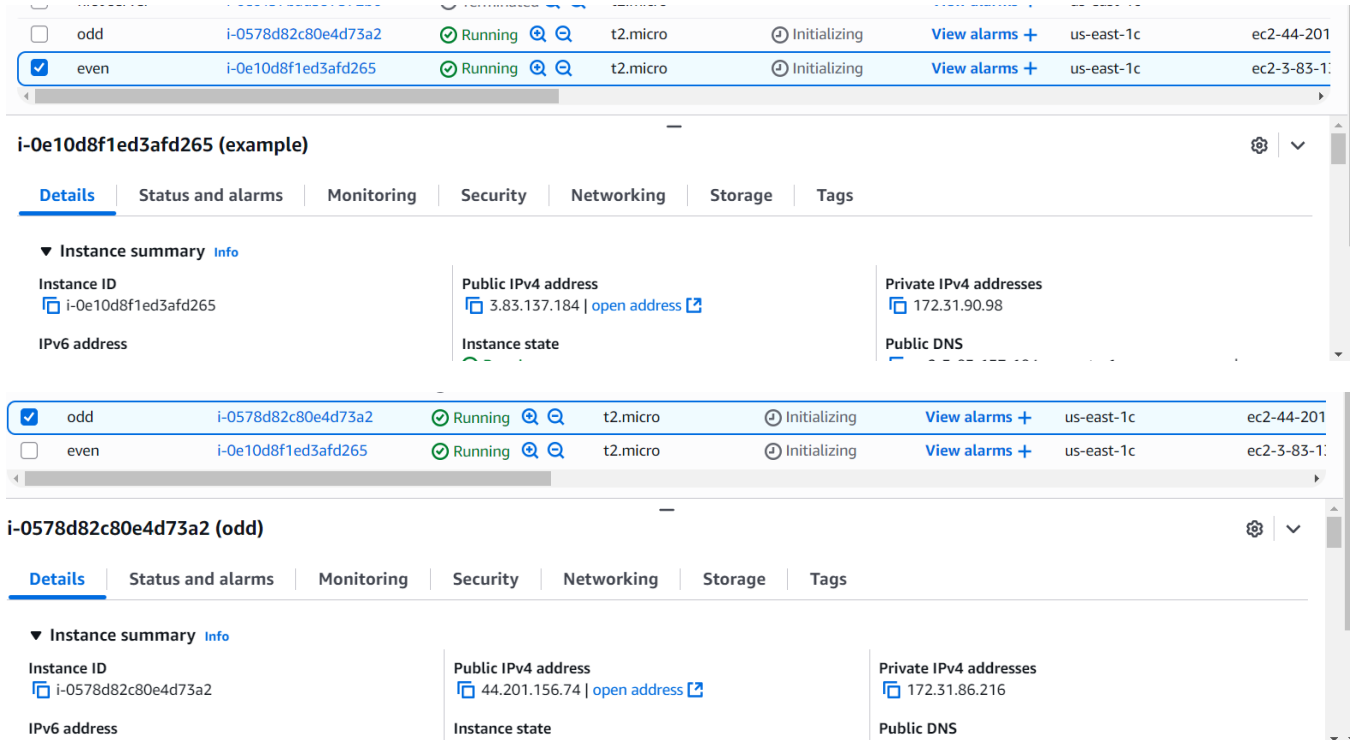


Building a Scalable Web Setup with AWS ALB and EC2

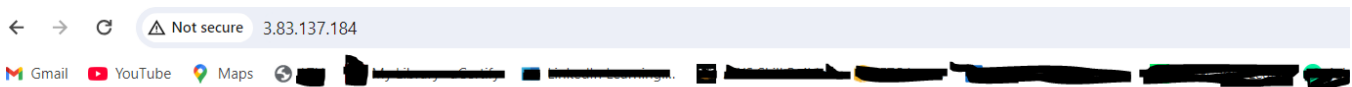
Hands-on project to learn ALB-based routing, EC2 setup, and security best practices on AWS.

Step1: launching two ec2 servers and checking their status by using public ip address



Instance ID	Public IPv4 address	Private IPv4 addresses
i-0578d82c80e4d73a2	44.201.156.74	172.31.86.216
i-0e10d8f1ed3afd265	3.83.137.184	172.31.90.98

Checking the working of instances by public ip



Hello welcome to aws ec2 instance ip-172-31-90-98.ec2.internal



Hello welcome to aws ec2 instance ip-172-31-86-216.ec2.internal

As we can see from the screenshots, that two ec2 instances had two different ip addresses to access them or the application. Our goal is to have one ip address for their two servers to access them and to balance the load between the two servers with the help of Application Load balancer.

Step 2: create ALB (Application load balancer) handles traffic of HTTP and HTTPS

Define security group: inbound HTTP from anywhere and outbound: anywhere

Create Target Group for EC2 instances:

VIEW
Review the load balancer configurations and make changes if needed. After you finish reviewing the configurations, choose **Create load balancer**.

Summary

Review and confirm your configurations. [Estimate cost](#)

Basic configuration [Edit](#)
Name: LOAD-BALANCER-ALB
Scheme: Internet-facing
IP address type: IPv4

Network mapping [Edit](#)
VPC: [vpc-05fa39d8c019631cb](#)
Public IPv4 IPAM pool: -
Availability Zones and subnets:

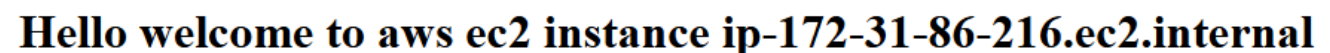
- us-east-1a
[subnet-0d2a5f918be0fc28](#)
- us-east-1b
[subnet-06c1d6a04a3b69efa](#)
- us-east-1c
[subnet-0e5daa7e56387510e](#)
- us-east-1d
[subnet-06701fcd92a3b582](#)
- us-east-1e
[subnet-0aaa8ca46ea4e4f53](#)
- us-east-1f
[subnet-06db88ac24f4fa1ff](#)

Security groups [Edit](#)
SG for load balancer ALB
[sg-0e7820af50ead77e8](#)

Listeners and routing [Edit](#)
HTTP:80 | Target group: [TG-for-ALB](#)

<input type="checkbox"/>	Name	DNS name	State	VPC ID	Availability Zones	Type
<input type="checkbox"/>	LOAD-BALANCER-ALB	LOAD-BALANCER-ALB-157...	Active	vpc-05fa39d8c019631cb	6 Availability Zones	application

I also got a DNS name for my load balancer as single point of Access and it is accessible too. We can see below of accessing two servers with one DNS name. Therefore our ALB is distributing load between them.



Step 3: now check the target group, it checks the health of each ec2 server by sending health checks now we will see the working of it by one server.

Let us stop the server of: 172.31.86.216

<input checked="" type="checkbox"/>	odd	i-0578d82c80e4d73a2	Stopping	t2.micro	Initializing	View alarms +	us-east-1c
<input type="checkbox"/>	even	i-0e10d8f1ed3afd265	Running	t2.micro	Initializing	View alarms +	us-east-1c

i-0578d82c80e4d73a2 (odd)

[Details](#) | [Status and alarms](#) | [Monitoring](#) | [Security](#) | [Networking](#) | [Storage](#) | [Tags](#)

▼ Instance summary [Info](#)

Instance ID i-0578d82c80e4d73a2	Public IPv4 address 44.201.156.74 open address	Private IPv4 addresses 172.31.86.216
------------------------------------	---	---

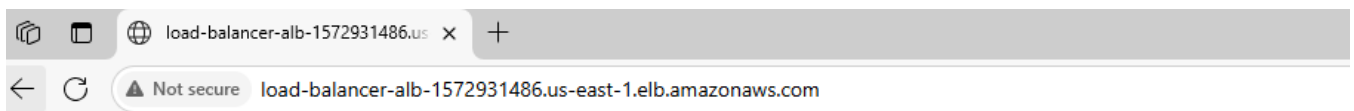
We can able to access only one server :

[Targets](#) | [Monitoring](#) | [Health checks](#) | [Attributes](#) | [Tags](#)

Registered targets (2) [Info](#) Anomaly mitigation: **Not applicable** [Deregister](#)

Target groups route requests to individual registered targets using the protocol and port number specified. Health checks are performed on all registered targets in a target group's health check settings. Anomaly detection is automatically applied to HTTP/HTTPS target groups with at least 3 healthy targets.

<input type="checkbox"/>	Instance ID	Name	Port	Zone	Health status	Health status details
<input type="checkbox"/>	i-0578d82c80e4d73a2	odd	80	us-east-1c (use...)	Unused	Target is in the stoppe...
<input type="checkbox"/>	i-0e10d8f1ed3afd265	even	80	us-east-1c (use...)	Healthy	-



Hello welcome to aws ec2 instance ip-172-31-90-98.ec2.internal

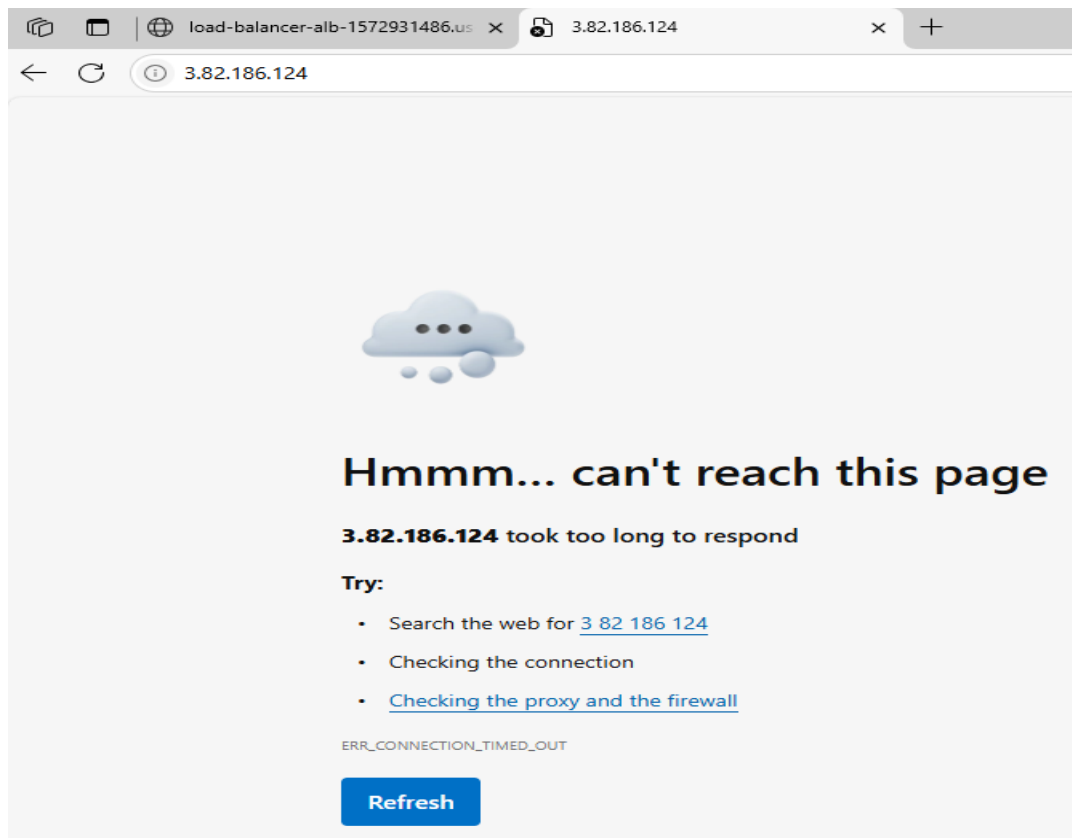
If I recover the stooped server and I can able access to both servers.

One improvement feature : I can able to access the servers from ALB DNS URL and also EC2 instance public IP, but this is not recommended in terms of security, we need to access the servers only from ALB DNS url only not from servers individual public ip address. Let us implement it.

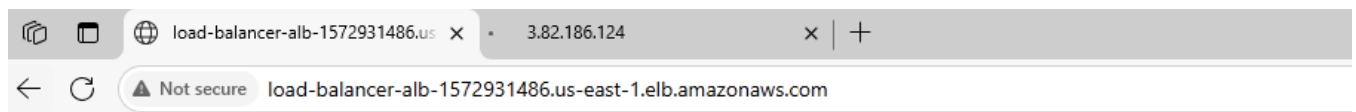
Step 1: change the security group of ec2 instances as its inbound rule should only allow traffic from ALB only

Inbound rules	Outbound rules	Sharing - new	VPC associations - new	Tags
Inbound rules (1)				
<input type="text" value="Search"/>				
Security group rule ID	IP version	Type	Protocol	Port range
sg-0ef93f02f482963ae	-	HTTP	TCP	80
				sg-0e7820af50ead77e8

Now try to access from servers public ip, and we should not able to access it



But if we access through DNS URL we can able see the working of ALB that forwards traffic for both the servers.



Hello welcome to aws ec2 instance ip-172-31-86-216.ec2.internal



And we achieved preventing traffic directly accessing our servers, now only through ALB traffic can able to reach servers.

One more additional feature we can use in ALB is to edit its rules as of now the current rule define that the traffic should goes to the servers and now we are going to add a rule for error url as path condition, also we can to define the rules on various conditions such as path ,Query string , source ip, http request mode etc

▼ Rule details

Priority 5	Conditions (If) If request matches all: Path Pattern is /error	Actions (Then) Return fixed response <ul style="list-style-type: none">Response code: 404Response body: Not found , custom errorResponse content type: text/plain
----------------------	--	--

Listener rules (2) [Info](#) [Rule limits](#) [Actions](#) [Add rule](#)

Traffic received by the listener is routed according to the default action and any additional rules. Rules are evaluated in priority order from the lowest value to the highest value.

<input type="checkbox"/>	Name tag	Priority	Conditions (If)	Actions (Then)	ARN	Ti
<input type="checkbox"/>	-	5	Path Pattern is /error	Return fixed response <ul style="list-style-type: none">Response code: 404Response body: Not found , custom errorResponse content type: text/plain	ARN	0...
<input type="checkbox"/>	Default	Last (default)	<i>If no other rule applies</i>	Forward to target group <ul style="list-style-type: none">TG-for-ALB: 1 (100%)Target group stickiness: Off	ARN	0...

We have two rules one is default that allows traffic to servers and the other one is error rule with least priority as 5.

Now we will check with /error:

