

LOAD BALANCER

A **load balancer** in AWS (Amazon Web Services) distributes incoming traffic across multiple targets (like EC2 instances, containers, and IP addresses) to ensure that no single instance or service is overwhelmed by traffic. AWS provides fully-managed load balancers through the **Elastic Load Balancing (ELB)** service, which can automatically scale to meet incoming traffic demands.

Types of Load Balancer

1. Application Load Balancer (ALB)

- Operates at the **Application Layer (Layer 7)** of the OSI model.
- Best suited for HTTP and HTTPS traffic. It is ideal for microservices and container-based applications (e.g., using ECS or Kubernetes).

Features:

- Supports host-based and path-based routing.
- Supports routing based on URL query strings and headers.
- Provides WebSockets and HTTP/2 support.
- Integrates with AWS Web Application Firewall (WAF) to help protect against common web threats.
- Can route traffic to multiple services based on rules, making it highly flexible.

2. Network Load Balancer (NLB)

- Operates at the **Transport Layer (Layer 4)**.
- Best for extreme performance, low-latency traffic, or when handling non-HTTP/S protocols (e.g., TCP, UDP).

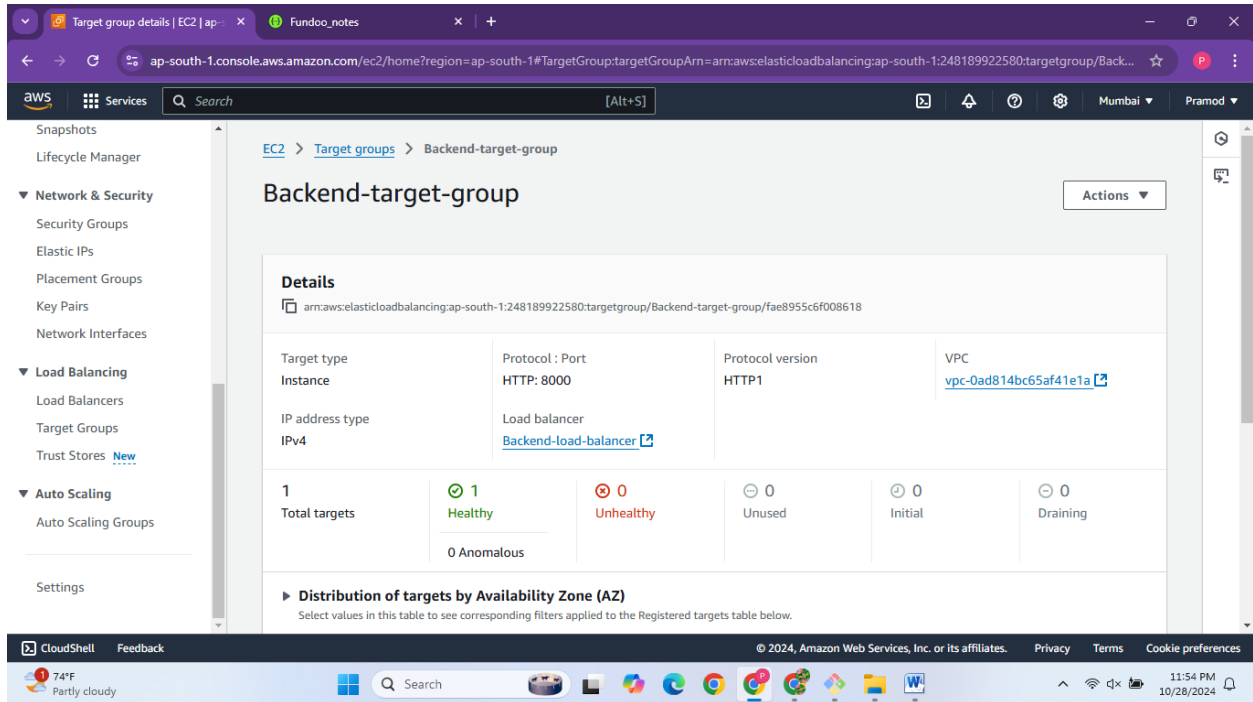
Features:

- Capable of handling millions of requests per second with very low latencies.
- Ideal for use cases where performance is critical (e.g., gaming, IoT).
- Supports static IP addresses, and Elastic IPs can be assigned to the load balancer.
- Preserves the source IP of the client, enabling IP-based targeting at the backend.

Implementation of Load Balancer for Fundoo Application

Load Balancer on Backend

Created Target group as 'Backend-target-group'



Protocol: HTTP

Port: 8000

Created Application Load Balancer as 'Backend-load-balancer'

Scheme: internal

Load balancer IP address type: IPv4

Availability zones: we have select minimum two different Availability zones and one subnet for one zone

- ap-south-1b (aps1-az3) – private-sub01
- ap-south-1c (aps1-az2) –private-subnet3

Security groups: I have selected Security group which I have created for my backend instances

- Backend-security

Listeners and routing:

Protocol: HTTP

Port: 80

Selected the target group that I have created:

- Backend-target-group

Then in target groups, we need to do health check for target group for Backend target group, whether it is working properly or not. If everything is correct then Health status will be 'Healthy' otherwise 'Unhealthy'.

In my frontend server, I need to curl the Backend server using DNS name of Backend Load Balancer.

I have added health check url in the urls.py file

Frontend Load Balancer

Created Target group as 'Frontend-target-group'

The screenshot displays the AWS Management Console interface for the 'Frontend-target-group'. The breadcrumb navigation shows 'EC2 > Target groups > Frontend-target-group'. The main content area is titled 'Frontend-target-group' and includes an 'Actions' dropdown menu. Below the title, the 'Details' section shows the ARN: `arn:aws:elasticloadbalancing:ap-south-1:248189922580:targetgroup/Frontend-target-group/b91dc5ab4bcc61d1`. A table lists the target group's configuration:

Target type	Protocol : Port	Protocol version	VPC
Instance	HTTP: 80	HTTP1	vpc-0ad814bc65af41e1a
IP address type	Load balancer		
IPv4	Frontend-load-balancer		

Below the configuration table, a summary row shows the target group's status:

Total targets	Healthy	Unhealthy	Unused	Initial	Draining
1	1	0	0	0	0

Below the summary row, a section titled 'Distribution of targets by Availability Zone (AZ)' indicates that 0 targets are anomalous.

The bottom of the screenshot shows the Windows taskbar with the date and time as 12:21 AM on 10/29/2024.

This will route traffic

Protocol: HTTP

Port: 80

Created Application Load Balancer as ‘Frontend-load-balancer’

Scheme: internet-facing

Load balancer IP address type: IPv4

.
. .

Health check path: /health/

Availability Zones:

ap-south-1a (aps1-az3) – public-sub01

ap-south-1b (aps1-az3) – public-sub03

Security groups: I have selected Security group which I have created for my backend instances

- Security group-project1

Listeners and routing:

Protocol: HTTP

Port: 80

Selected the target group that I have created:

- frontend-target-group

Then in target groups, we need to do health check for target group for frontend, whether it is working properly or not. If everything is correct then Health status will be ‘Healthy’ otherwise ‘Unhealthy’.

In frontend server, we need to add the DNS name of Backend Load Balancer instead of ip address of Backend instance in the fundoo.conf file

```
server {  
    listen 80;  
    server_name _default;  
location / {  
    include proxy_params;  
proxy_pass http://internal-Backend-load-balancer-1061140472.ap-south-  
1.elb.amazonaws.com:8000;  
}  
}
```

In browser, DNS name of Frontend Load Balancer should get the application Fundoo.