LOAD BALANCER

Introduction

AWS load balancers are a critical component of Amazon Web Services (AWS) that distribute incoming application traffic to multiple targets. This can improve the performance, reliability, and availability of applications hosted on AWS

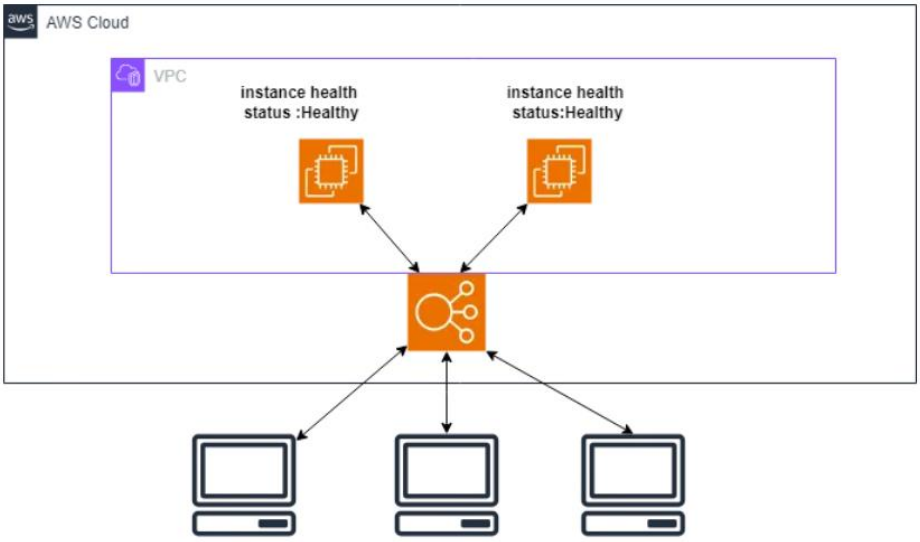
What is Elastic Load Balancing?

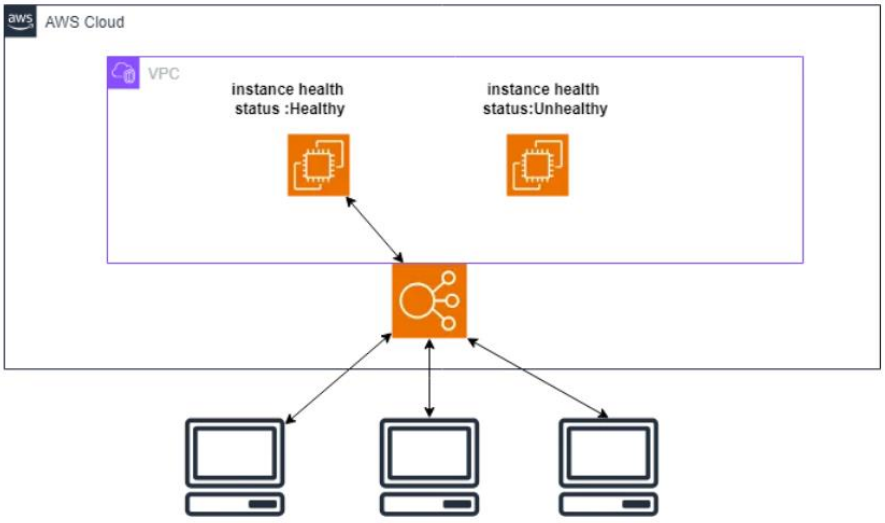
- ELB is a service that automatically distributes incoming application traffic and scale resources to meet traffic demands.
- ELB helps in adjusting capacity according to incoming application and network traffic.
- ELB can be enabled within a single availability zone or across multiple availability zones to maintain consistent application performance.

What are ELB features?

- Detection of unhealthy EC2 instances.
- Spreading EC2 instances across healthy channels only.
- Centralized management of SSL certificates.
- Optional public key authentication.
- Support for both IPv4 and IPv6.
- ELB accepts incoming traffic from clients and routes requests to its registered targets.
- When an unhealthy target or instance is detected, ELB stops routing traffic to it and resumes only when the instance is healthy again.
- ELB monitors the health of its registered targets and ensures that the traffic is routed only to healthy instances.
- ELB's are configured to accept incoming traffic by specifying one or more listeners. A listener is a process that checks for connection requests.
- Listeners are configured with a protocol and port number from the client to the ELB and vice-versa i.e., back from ELB to the client.

Architecture Diagram





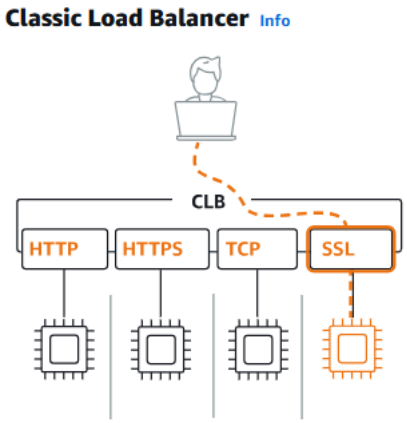
What are ELB types of load balancers ?

- Classic Load Balancer
- Application Load Balancer
- Network Load Balancer
- Gateway Load Balancer[Available in specific regions only]

Classic Load balancer

A Classic Load Balancer (CLB) is one of the earlier versions of load balancers provided by AWS under the Elastic Load Balancing (ELB) service.

If we have equal numbers of instances in both AZ then classic can be used.



Features of Classic Load Balancer:

Layer 4 and Layer 7 Support:

- Distributes traffic based on TCP/SSL (Layer 4) and HTTP/HTTPS (Layer 7) protocols.
- Can inspect and route HTTP headers for Layer 7 load balancing.

Basic Routing Capabilities:

- Routes traffic to EC2 instances based on their health and availability.
- No advanced features like content-based routing or host/path-based routing (available in ALBs).

Health Checks:

- Monitors the health of targets using TCP connections, HTTP requests, or HTTPS requests.
- Redirects traffic away from unhealthy instances.

SSL Termination:

- Can handle SSL certificates for HTTPS traffic, offloading the SSL processing from backend instances.

Static IPs:

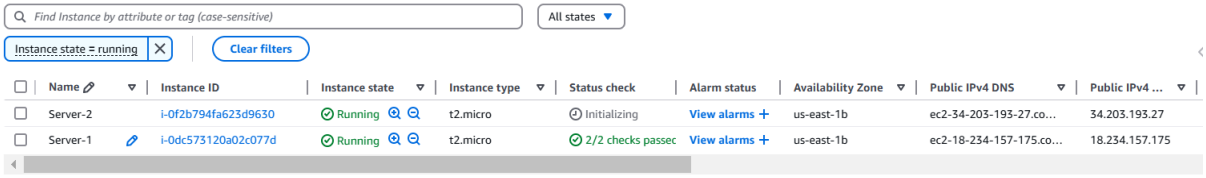
- Does not support static IP addresses. Instead, it uses domain names that resolve to dynamic IPs.

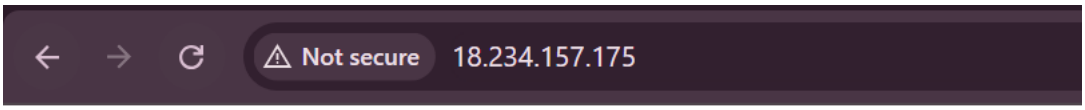
Limited Scaling Capabilities:

- Automatically scales with the traffic load but has limitations compared to ALB or NLB in handling modern web application traffic.

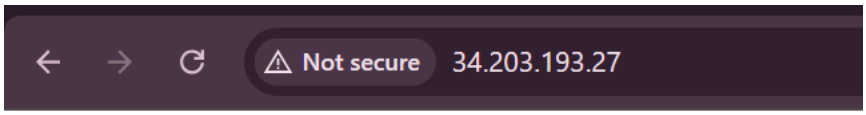
EXECUTION

Launch the 2 ec2 instances with the httpd user data





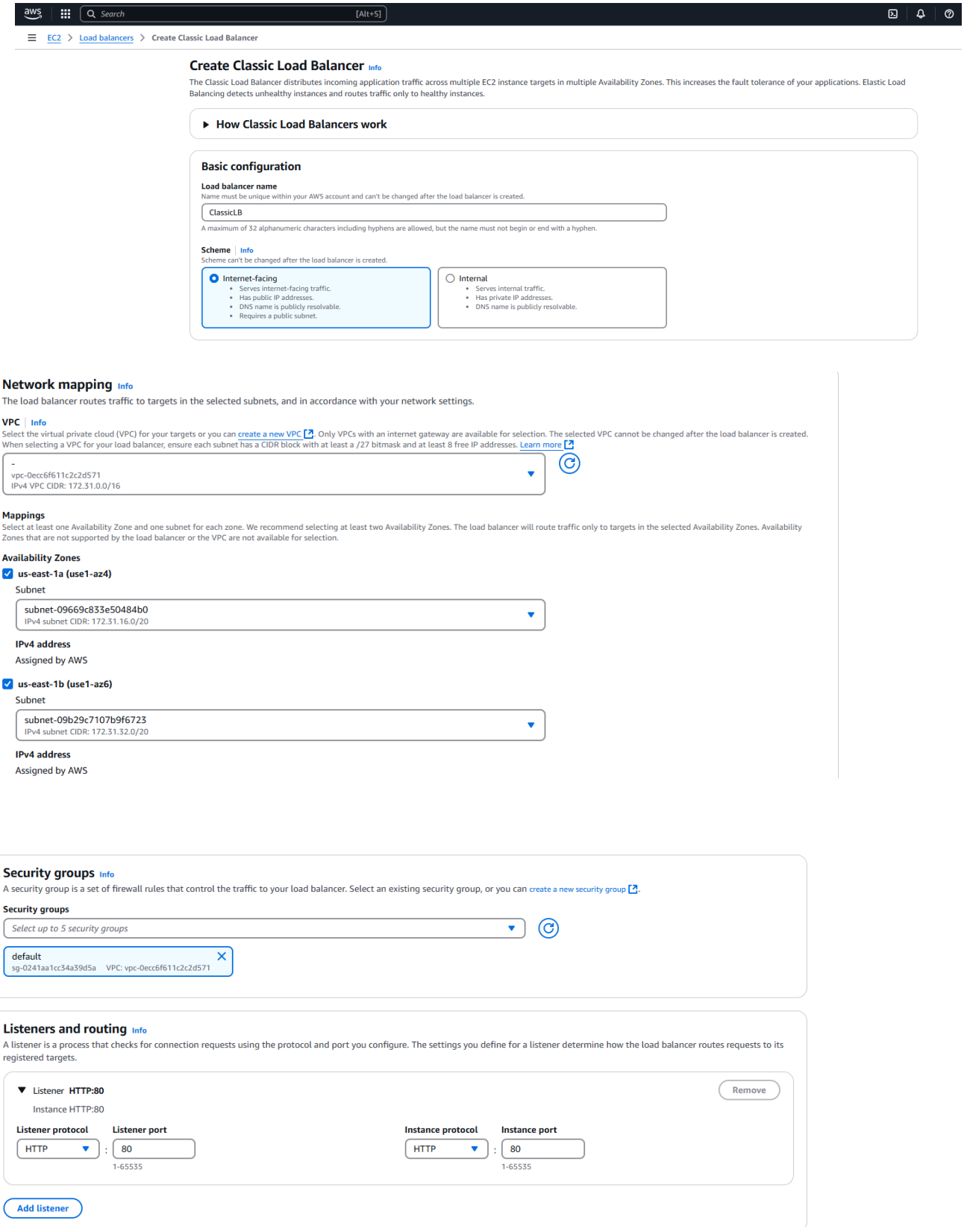
Welcome to the aws

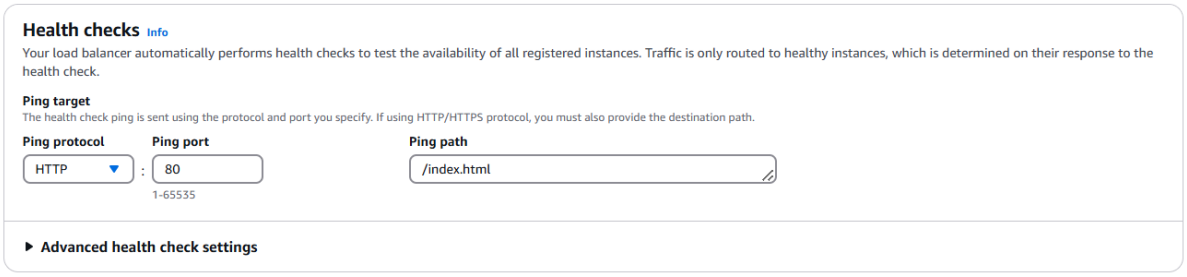


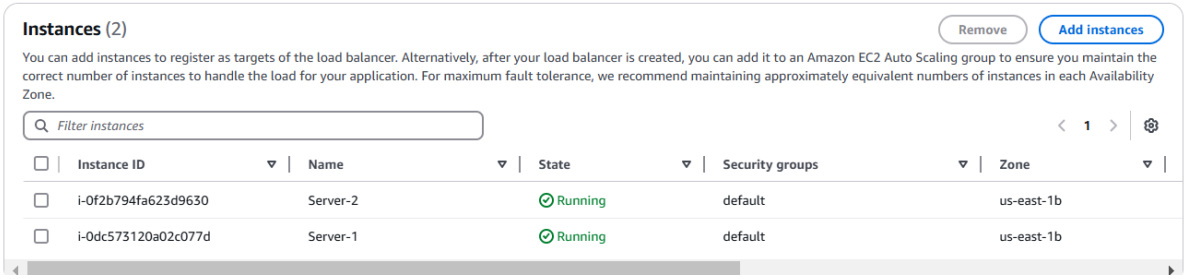
Welcome to aws[2]

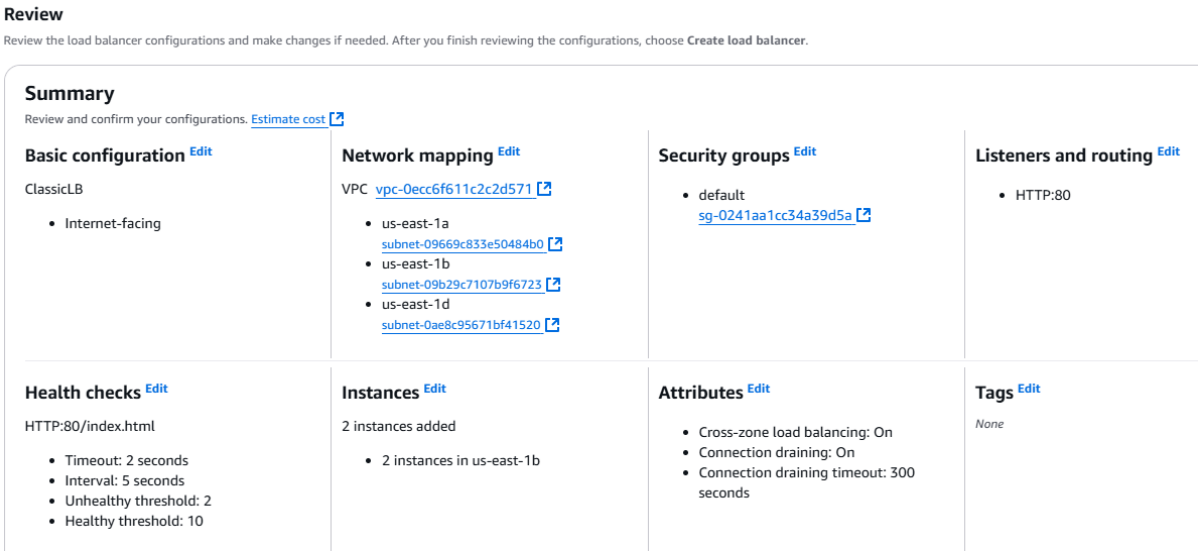
Now creating the classic Load balancer

- Go to the Loadbalancer[ec2 feature] in aws
- Select Classic loadbalancer

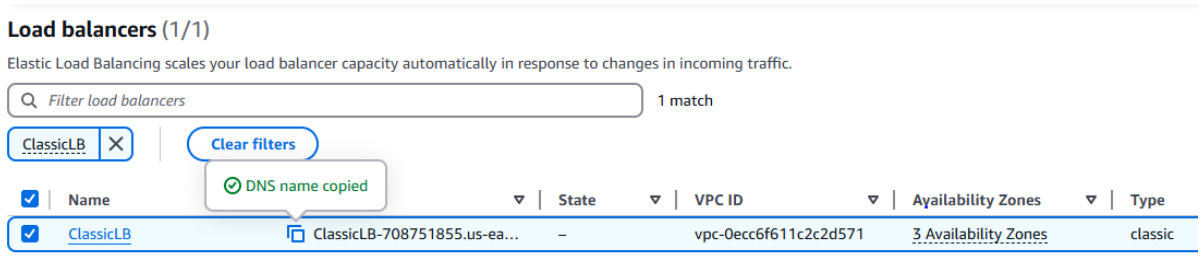

aws
Search
[Alt+S]
EC2 > Load balancers > Create Classic Load Balancer
Create Classic Load Balancer Info
The Classic Load Balancer distributes incoming application traffic across multiple EC2 instance targets in multiple Availability Zones. This increases the fault tolerance of your applications. Elastic Load Balancing detects unhealthy instances and routes traffic only to healthy instances.
How Classic Load Balancers work
Basic configuration
Load balancer name
Name must be unique within your AWS account and can't be changed after the load balancer is created.
ClassicLB
A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.
Scheme Info
Scheme can't be changed after the load balancer is created.
Internet-facing
Serves internet-facing traffic.
Has public IP addresses.
DNS name is publicly resolvable.
Requires a public subnet.
Internal
Serves internal traffic.
Has private IP addresses.
DNS name is publicly resolvable.
Network mapping Info
The load balancer routes traffic to targets in the selected subnets, and in accordance with your network settings.
VPC Info
Select the virtual private cloud (VPC) for your targets or you can create a new VPC. Only VPCs with an internet gateway are available for selection. The selected VPC cannot be changed after the load balancer is created.
When selecting a VPC for your load balancer, ensure each subnet has a CIDR block with at least a /27 bitmask and at least 8 free IP addresses. Learn more
-
vpc-0ecc6f611c2c2d571
IPv4 VPC CIDR: 172.31.0.0/16
Mappings
Select at least one Availability Zone and one subnet for each zone. We recommend selecting at least two Availability Zones. The load balancer will route traffic only to targets in the selected Availability Zones. Availability Zones that are not supported by the load balancer or the VPC are not available for selection.
Availability Zones
us-east-1a (use1-az4)
Subnet
subnet-09669c833e50484b0
IPv4 subnet CIDR: 172.31.16.0/20
IPv4 address
Assigned by AWS
us-east-1b (use1-az6)
Subnet
subnet-09b29c7107b9f6723
IPv4 subnet CIDR: 172.31.32.0/20
IPv4 address
Assigned by AWS
Security groups Info
A security group is a set of firewall rules that control the traffic to your load balancer. Select an existing security group, or you can create a new security group.
Security groups
Select up to 5 security groups
default
sg-0241aa1cc34a39d5a VPC: vpc-0ecc6f611c2c2d571
Listeners and routing Info
A listener is a process that checks for connection requests using the protocol and port you configure. The settings you define for a listener determine how the load balancer routes requests to its registered targets.
Listener HTTP:80
Remove
Instance HTTP:80
Listener protocol
Listener port
HTTP
80
1-65535
Instance protocol
Instance port
HTTP
80
1-65535
Add listener



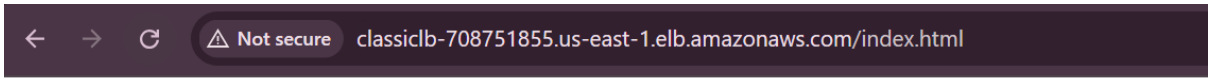




Next click “ create”

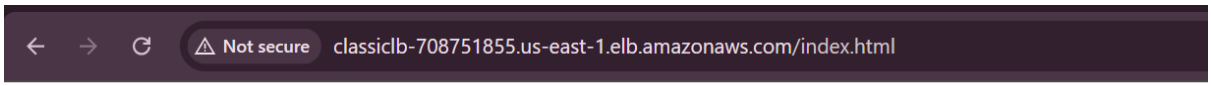


Copy the DNS name and paste it on the browser



Welcome to the aws

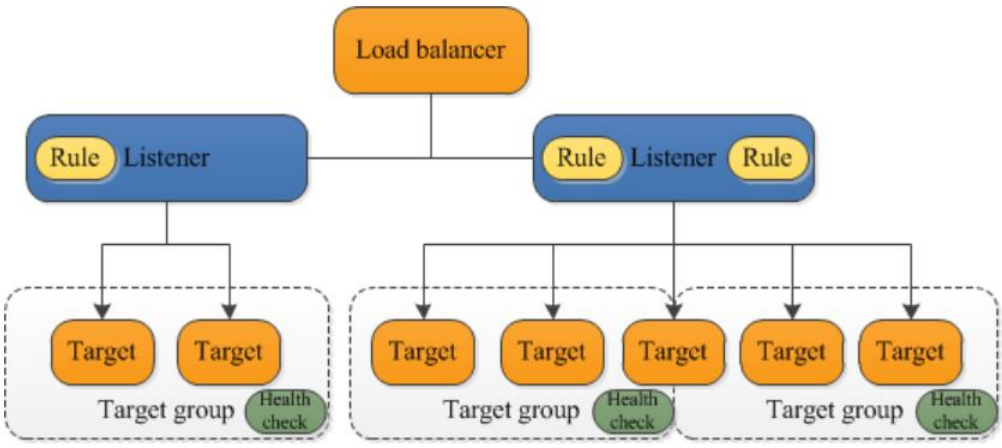
Refresh the browser



Welcome to aws[2]

APPLICATION LOAD BALANCER

- Application Load Balancer operates at layer 7 (application layer) and allows defining routing rules based on content across multiple services or containers running on one or more EC2 instances.
- scales the load balancer as traffic to the application changes over time.
- can scale to the vast majority of workloads automatically.
- supports health checks, used to monitor the health of registered targets so that the load balancer can send requests only to the healthy targets.



An Application Load Balancer functions at the application layer, the seventh layer of the Open Systems Interconnection (OSI) model.

After the load balancer receives a request, it evaluates the listener rules in priority order to determine which rule to apply, and then selects a target from the target group for the rule action.

You can configure listener rules to route requests to different target groups based on the content of the application traffic. Routing is performed independently for each target group, even when a target is registered with multiple target groups.

You can configure the routing algorithm used at the target group level. The default routing algorithm is round robin; alternatively, you can specify the least outstanding requests routing algorithm.

You can add and remove targets from your load balancer as your needs change, without disrupting the overall flow of requests to your application. Elastic Load Balancing scales your load balancer as traffic to your application changes over time. Elastic Load Balancing can scale to the vast majority of workloads automatically.

You can configure health checks, which are used to monitor the health of the registered targets so that the load balancer can send requests only to the healthy targets.

Benefits of migrating from a Classic Load Balancer

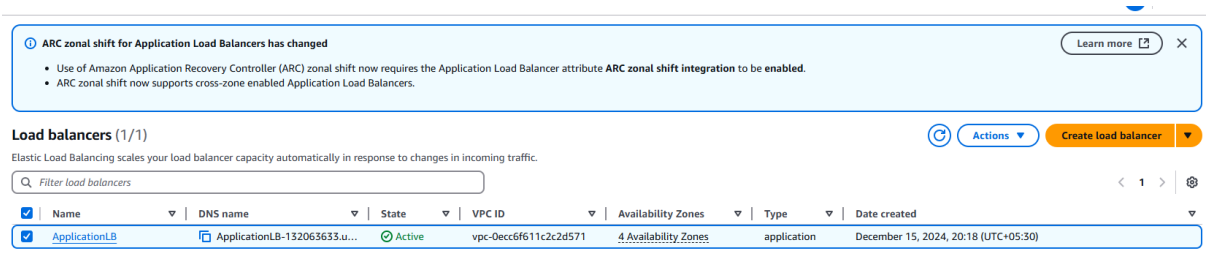
Using an Application Load Balancer instead of a Classic Load Balancer has the following benefits:

- Support for Path conditions. You can configure rules for your listener that forward requests based on the URL in the request. This enables you to structure your application as smaller services, and route requests to the correct service based on the content of the URL.
- Support for Host conditions. You can configure rules for your listener that forward requests based on the host field in the HTTP header. This enables you to route requests to multiple domains using a single load balancer.
- Support for routing based on fields in the request, such as HTTP header conditions and methods, query parameters, and source IP addresses.
- Support for routing requests to multiple applications on a single EC2 instance. You can register an instance or IP address with multiple target groups, each on a different port.
- Support for redirecting requests from one URL to another.

- Support for returning a custom HTTP response.
- Support for registering targets by IP address, including targets outside the VPC for the load balancer.
- Support for registering Lambda functions as targets.
- Support for the load balancer to authenticate users of your applications through their corporate or social identities before routing requests.
- Support for containerized applications. Amazon Elastic Container Service (Amazon ECS) can select an unused port when scheduling a task and register the task with a target group using this port. This enables you to make efficient use of your clusters.
- Support for monitoring the health of each service independently, as health checks are defined at the target group level and many CloudWatch metrics are reported at the target group level. Attaching a target group to an Auto Scaling group enables you to scale each service dynamically based on demand.
- Access logs contain additional information and are stored in compressed format.
- Improved load balancer performance.

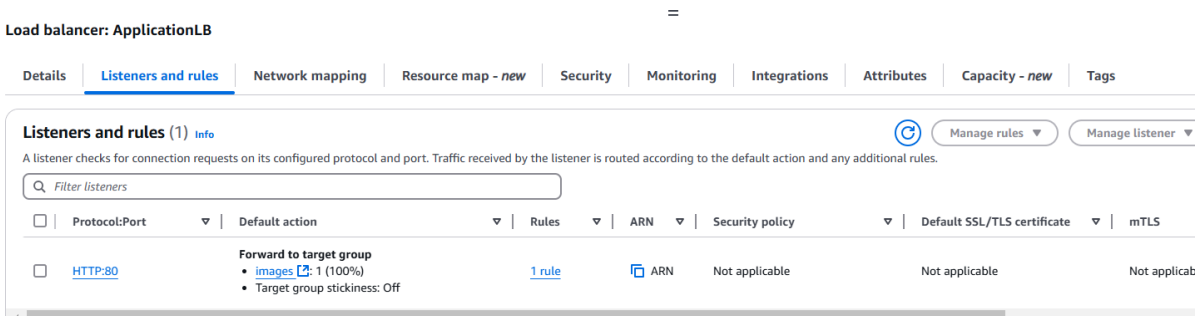
EXECUTION

Created an application load balancer



Attached target group

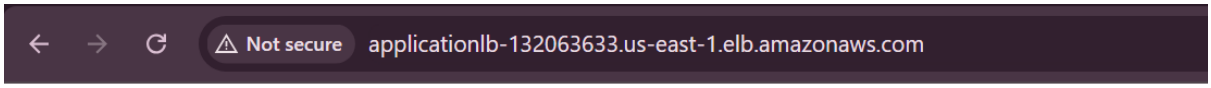
=



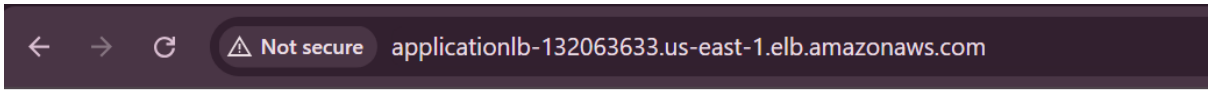
We can give path prefix also

```
[root@ip-172-31-45-138 html]# mkdir images
[root@ip-172-31-45-138 html]# ls
images  index.html
[root@ip-172-31-45-138 html]# cd images/
[root@ip-172-31-45-138 images]# vi index.html
[root@ip-172-31-45-138 images]#
```

Output

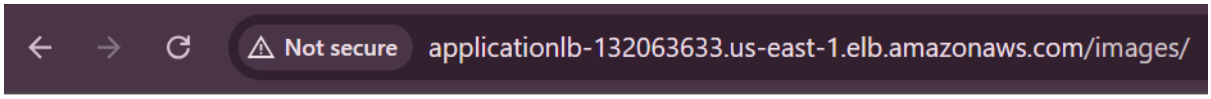


Welcome to aws[2]



Welcome to the aws

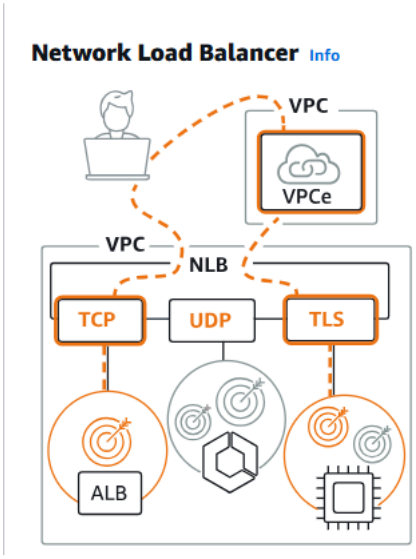
with the path prefix



Hello from Images!

NETWORK LOAD BALANCER

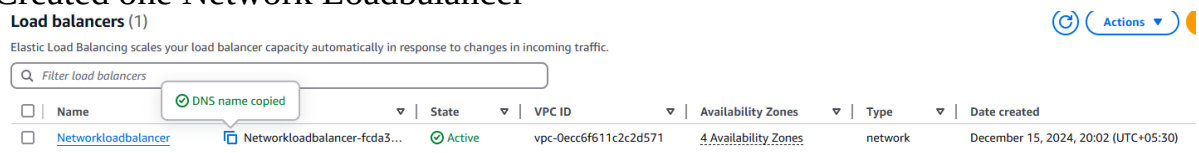
A Network Load Balancer functions at the fourth layer of the Open Systems Interconnection (OSI) model. It can handle millions of requests per second. After the load balancer receives a connection request, it selects a target from the target group for the default rule. It attempts to open a TCP connection to the selected target on the port specified in the listener configuration

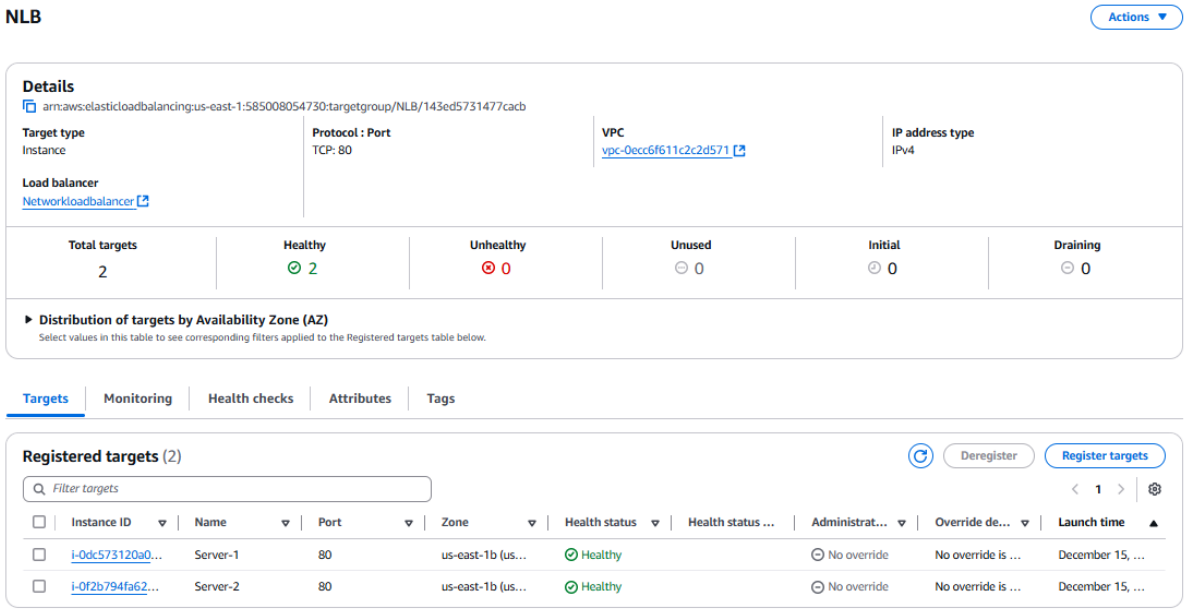


Network Load Balancer works on TCP/UDP Protocols
TCP/UDP Protocols works on the Transport layer that is Layer 4.
We can configure different Listeners.
It will work based on path prefix and rules configure .Target group will be attached to the load balancer, we can assign a elastic ip to NLB.
NLB will be used if we are not sure of hitting our application and if we are expecting sudden spikes in traffic then NLB can be used handle such situations.

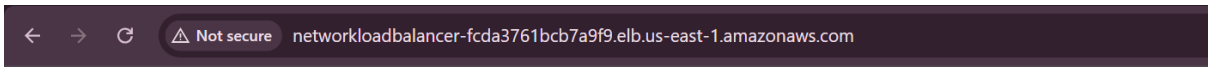
EXECUTION

Created one Network Loadbalancer

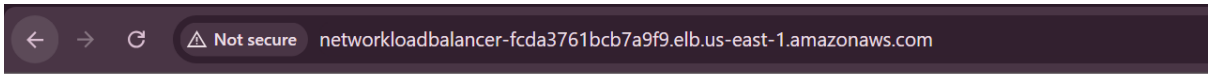




OUTPUT



Welcome to the aws



Welcome to aws[2]