

Elastic Load Balancing automatically distributes user incoming traffic across multiple targets, such as EC2 instances, containers, and IP addresses, in one or more Availability Zones.

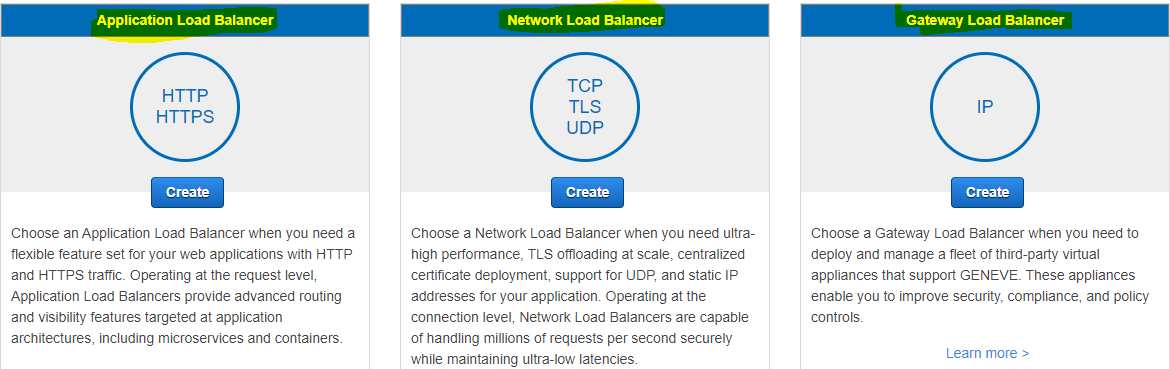
It monitors the health of its registered targets, and routes traffic only to the healthy targets.

**Types of Load balancers:**

**1) Application**

**2) Network**

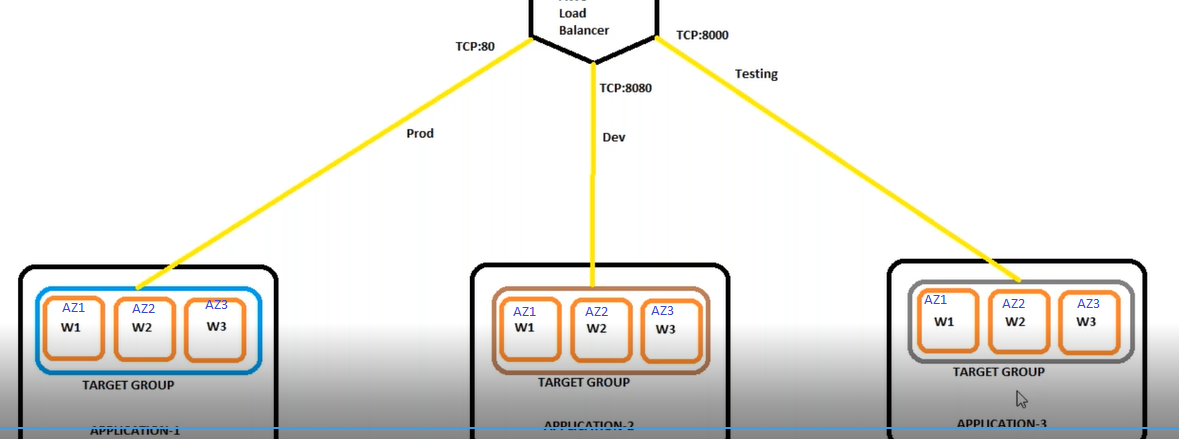
**3) Gateway**



**Network Load Balancer:**

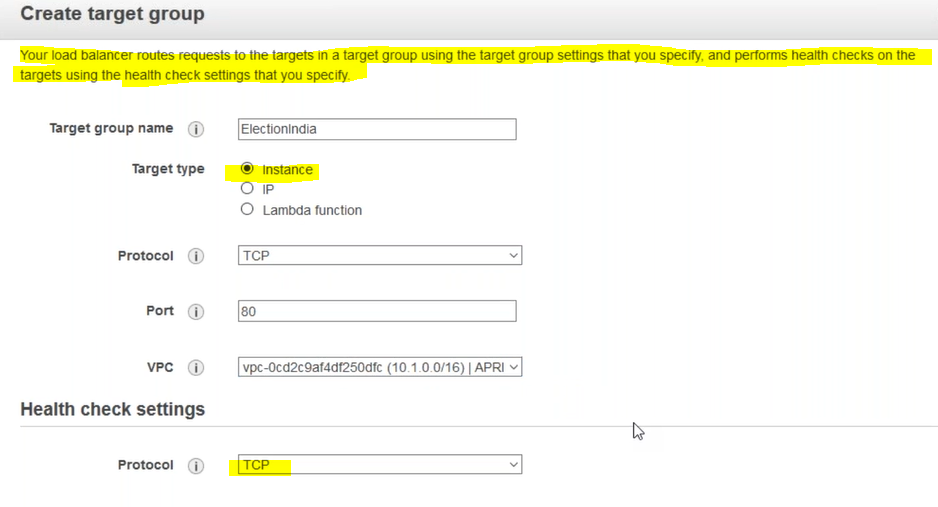
* A Network Load Balancer functions at the fourth layer(Transport) of the Open Systems Interconnection (OSI) model.
* The Network load balancer simply distributes incoming traffic across multiple targets, such as Amazon EC2 instances.
* A *listener* checks for connection requests from clients, using the protocol and port that you configure, and forwards requests to a target group.

**Note** :: We can load balance multiple applications using single ELB

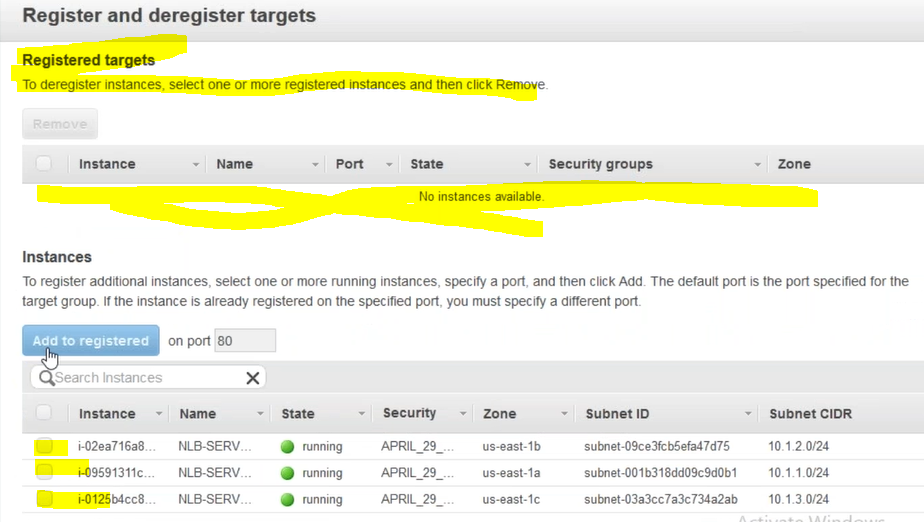


**Note** :: ELB is region specific

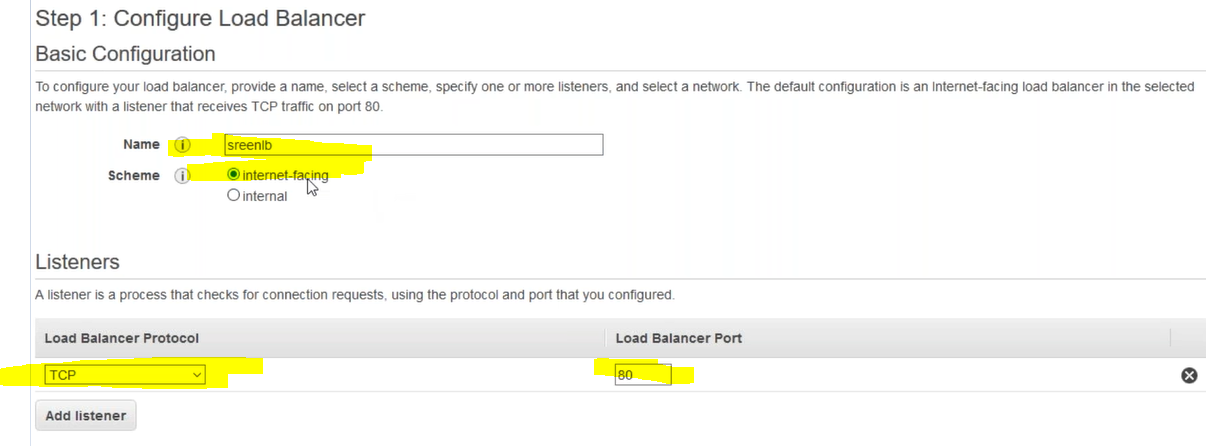
1. Create target group

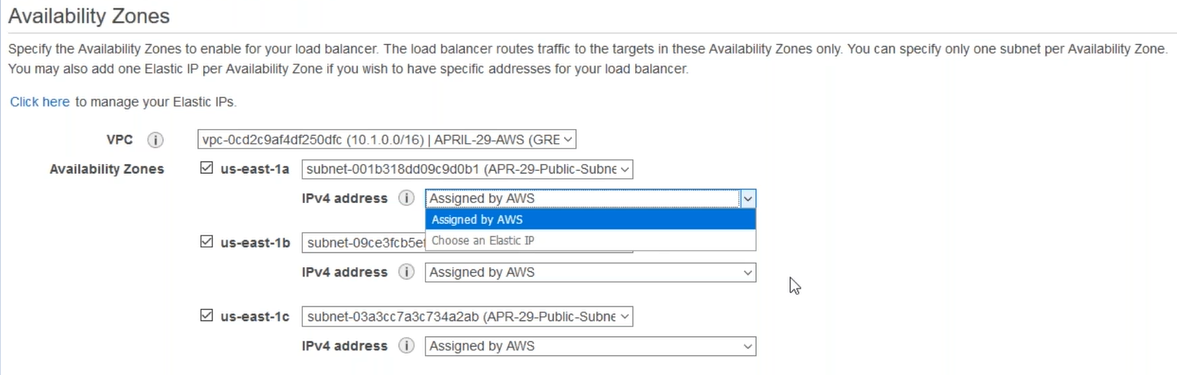


b) Add targets to the target group:



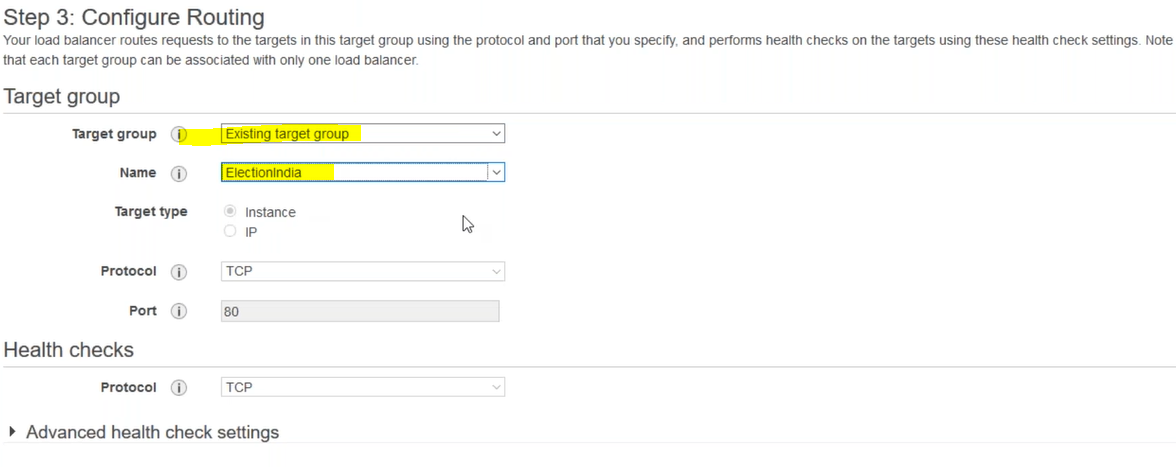
C) Create load balancer

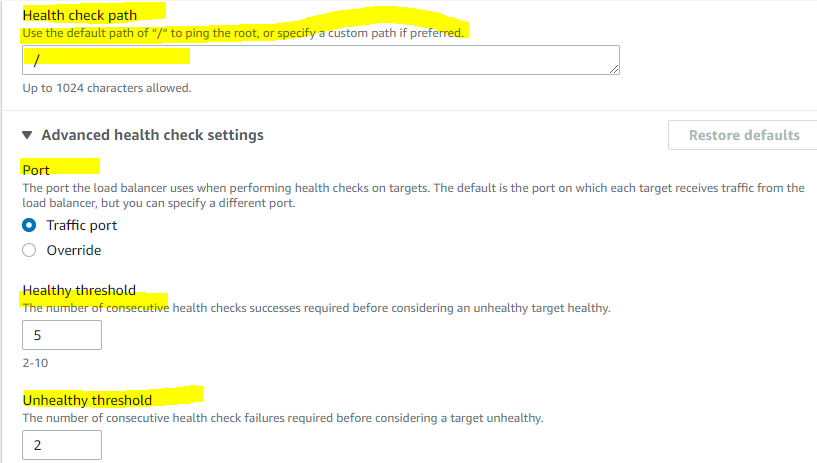


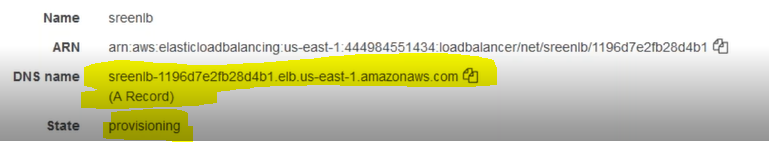


**Note** :: While creating NLB ,we can’t add availability zone after NLB creation(Need to add at the time of creation itself)

d) Select Target group

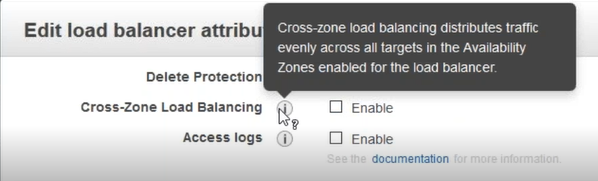




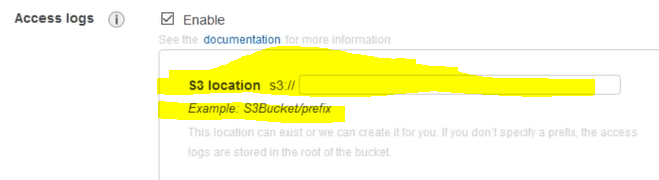


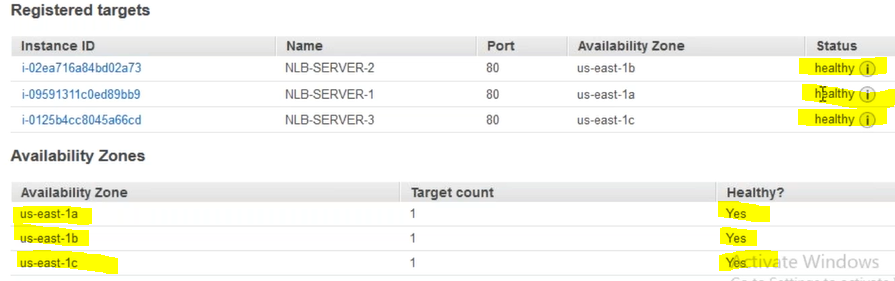


**What is cross-zone Load balancing:**

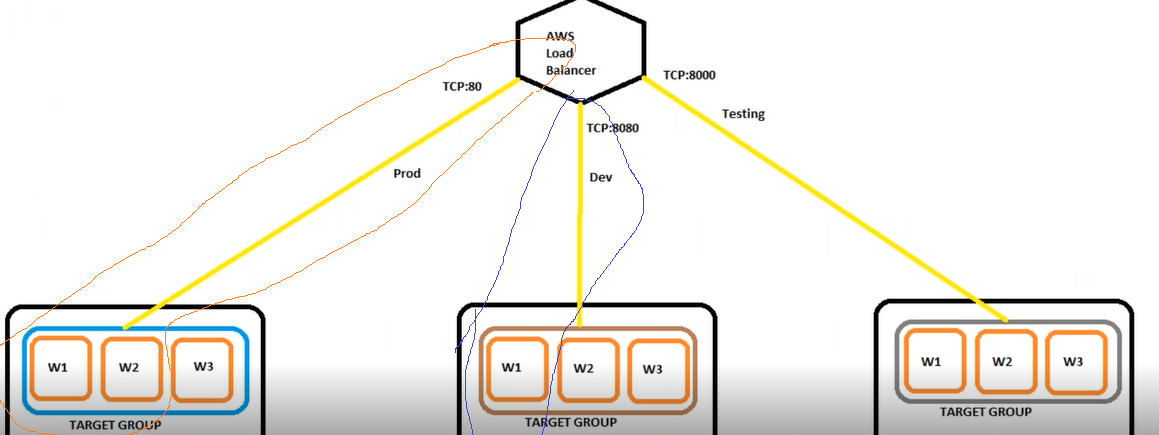


**We can also store logs to S3 bucket**:( Use case : if any user is unable to access application we can refer the logs by analysing through splunk or other tools)

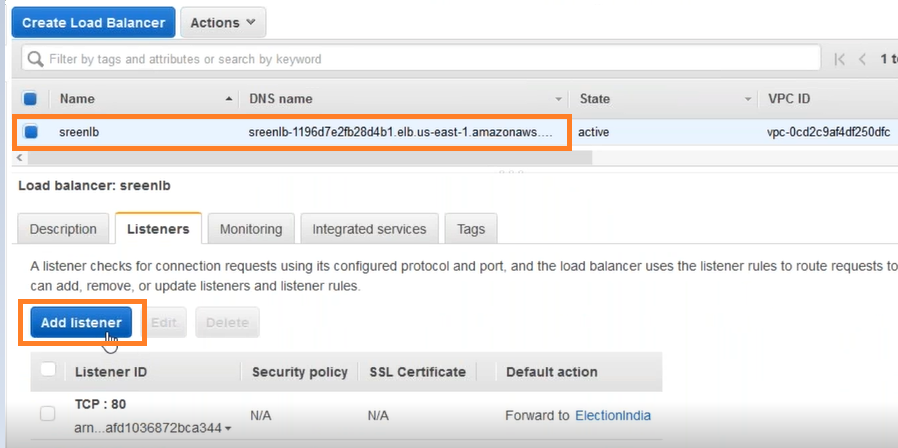


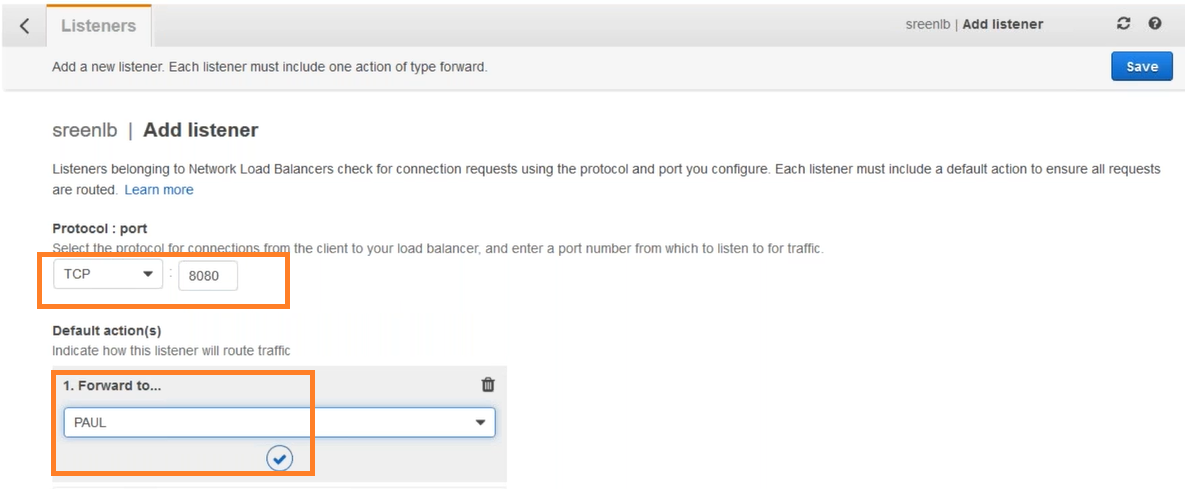


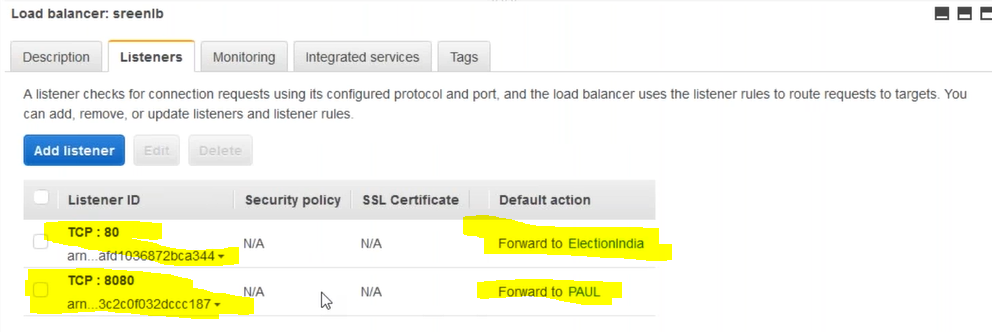
**When we want to add instance from diff target group to the same ELB ,chose diff port**



**Add listeners**





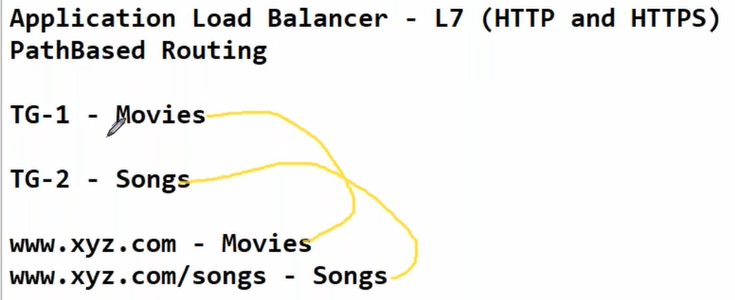


**Application Load Balancer (ALB):**

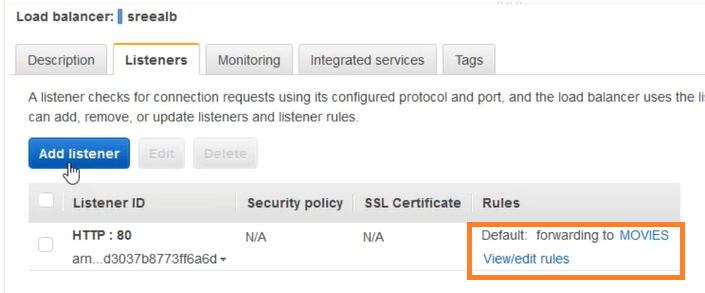
* An Application Load Balancer functions at the application layer(L7), the seventh layer of the Open Systems Interconnection (OSI) model.
* After the load balancer receives a request, it evaluates the listener rules in priority order to determine which rule to apply, and then selects a target from the target group for the rule action.

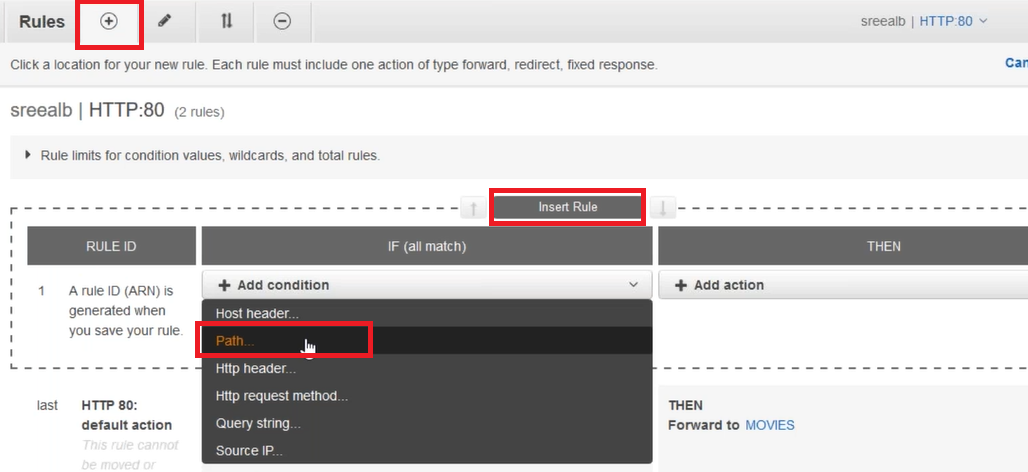
The following diagram illustrates the basic components. Notice that each listener contains a default rule, and one listener contains another rule that routes requests to a different target group. One target is registered with two target groups.

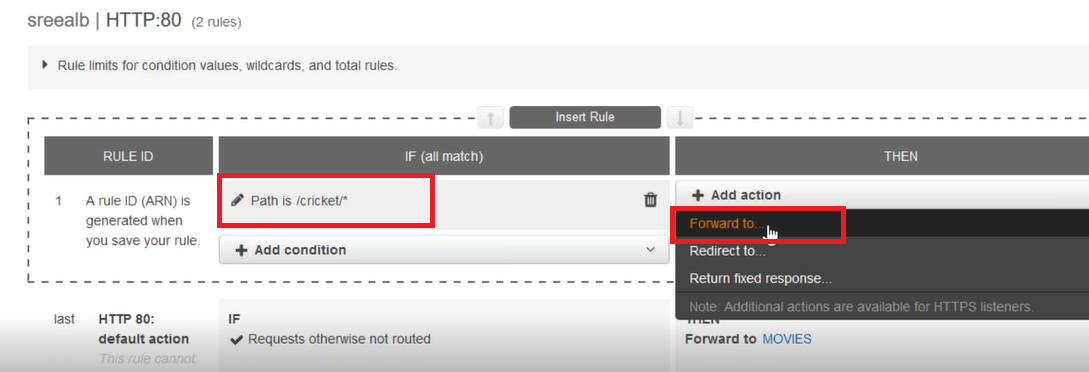

                The components of a basic Application Load Balancer
            



**Note ::**For ALB ,need to edit the rules ,no need to add listener as we did in case of NLB







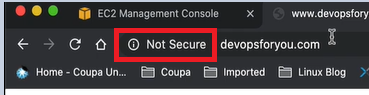


|  |  |
| --- | --- |
| **Network Load Balancer** | **Application Load Balancer** |
| Works at L4(Tranport) | Works at L7(Application) |
| Supports TCP,TLS and UDP | Supports HTTP,HTTPS |
| High performance | Low performance compared to NLB |
| x | Path/host based routing |
| x | Cross zone LB(distributes evenly) |
| x | Sticky session |
| Need to add listener ports | Need to add rules |
| Ones NLB is created we can’t add AZ again | In ALB we can add AZ’s anytime |
| **x** | SSL offloading/termination |
| Simply routes the traffic to the target group | it evaluates the listener rules, and then routes the traffic to the target group |

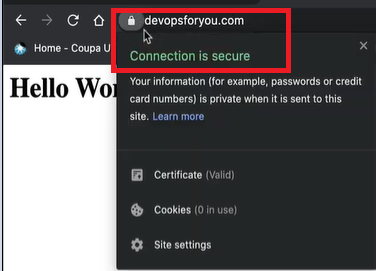
\*A sticky one keeps a user's session on the same server where it started. A non-sticky balancer can put each request in a session on a different server

\*SSL offloading is the process of removing the SSL-based encryption from incoming traffic to relieve a web server of the processing burden of decrypting and/or encrypting traffic sent via SSL. The processing is offloaded to a separate device designed specifically for [SSL acceleration](https://www.f5.com/services/resources/glossary/ssl-acceleration) or [SSL termination](https://www.f5.com/services/resources/glossary/ssl-termination).

Before:



After:



Host-based routing allows you to route a request based on host field in the HTTP header.

Path Based Routing allows you to route traffic to back-end server pools based on URL Paths of the request

**Note** :: 100rules and 5 conditions per rule

