Design and Implement Network Load Balancer with Multi Availability Zone using Amazon Web Server

Shahbaz Nadim Ansari[1], Farha Khan[2]

[1]Mittal Institute of Technology, Digital Communication Department, Bhopal, India
nadiiim25[at]gmail.com

[1]Mittal Institute of Technology, Digital Communication Department, Bhopal, India
farhakhan1609[at]gmail.com

Abstract: ***Cloud computing is the on-demand delivery of IT resources over the Internet with pay-as-you-go pricing. Instead of buying, owning, and maintaining physical data centers and servers, you can access technology services, such as computing power, storage, and databases, on an as-needed basis from a cloud provider like Amazon Web Services (AWS), Microsoft Azure, Google cloud. Cloud computing is one of the most growing technologies. The fundamental idea behind cloud computing is to distribute an array of computing services by unifying and scheduling a pool of computing resources, thereby minimizing the burden on the users and helping them focus on their core businesses. These computing resources are hosted on virtual hosts and distributed on-demand to the users by cloud service providers. For efficient resource utilization, systematic load balancing of incoming user traffic across virtual hosts is imperative. Aim of this paper is to design and implement Network load balancer with cross zone feature which balance incoming user requests and avoid the problem of server down in real scenario. Elastic Load Balancer automatically equally distribute incoming traffic to the available EC2, targets, like Amazon EC2 instances, IP addresses containersand Lambda functions. Load balancer effectively distribute load equally on available server and AZ to improve fault tolerance and availability. It can manage the differing load of your application traffic in one Availability Zone or over different Availability Zones.***

Keywords: Amazon web server, cloud computing, Load balancer, Availability zone, EC2 Instance

1. Introduction

Cloud Computing has emerged as a combination of distributed computing technology, server virtualization technology and network storage. The fundamental idea of cloud computing is to distribute an array of computing services by unifying and scheduling a pool of computing resources, thereby minimizing the burden on users and helping them focus on their core businesses [1] [2]. Cloud computing exhibits various characteristics such as scalability, location independence, flexibility, device independence, elasticity, reliability, resource sharing, cost effectiveness and on-demand computing [3]. All these factors have led to the accelerating use of cloud computing. A load balancing operation consists of three rules. These are location rule, distribution rule and selection rule [2, 5] The selection rule works either in pre-emptive or in non-preemptive fashion. The newly generated process is always picked up by the non-pre-emptive rule while the running process may be picked up by the preemptive rule. Pre-emptive transfer is costly than non-preemptive transfer which is more preferable. However preemptive transfer is more excellent than non-pre-emptive transfer in some instances. Practically load balancing decisions are taken jointly by location and distribution rules. The balancing domains are of two types: local and global. In local domain, the balancing decision is taken from a group of nearest neighbours by exchanging the local workload information while in global domain the balancing decision is taken by triggering. Load balancing plays a very important role in the networking technology, Load balancer comes into play when the user tries to connect to the server. This sample example is shown in the Fig 1 Any requests made to the internet will reach the server in various paths. There may be N number of servers which are serving the purpose of the request but the request will go to only one depending upon the traffic. The traffic here refers to how busy server is, in responding the requests made by servers. Requests will be directed towards servers by one of the networks called Load balancer which balances the load of server.

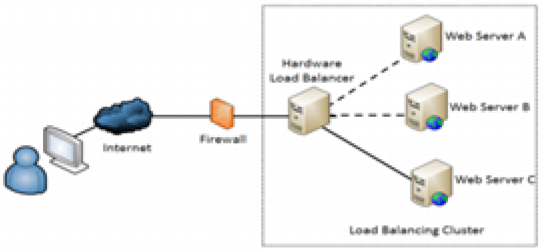


Figure 1: Load balancer with multi server

There are different types of load balancer which handles the traffic in different ways. Major types of Load balancer are Static and Dynamic Load balancer. In static algorithm the processes are assigned to the processors at the compile time according to the performance of the nodes. Once the processes are assigned, no change or reassignment is possible at the run time Number of jobs in each node is fixed in static load balancing algorithm. Static algorithms do not collect any information about the nodes The assignment of jobs is done to the processing nodes on the basis of the following factors: incoming time, extent of resource needed, mean execution time and inter-process communications. Since these factors should be measured before the assignment, this is why static load balance is also called probabilistic algorithm. As there is no migration of job at the runtime no overhead occurs or a little over head may occur. In static load balancing it is observed that as the number of tasks is more than the processors, better will be the load balancing. During the static load balancing too much information about the system and jobs must be known before the execution. This information may not be available in advance. A thorough study on the system state and the jobs quite tedious approach in advance. So, dynamic load balancing algorithm came into existence. The assignment of jobs is done at the runtime. In DLB jobs are reassigned at the runtime depending upon the situation that is the load will be transferred from heavily loaded nodes to the lightly loaded nodes. In this case communication over heads occur and becomes more when number of processors increase. dynamic load balancing no decision is taken until the process gets execution. This strategy collects the information about the system state and about the job information. As more information is collected by an algorithm in a short time, potentially the algorithm can make better decision. Dynamic load balancing is mostly considered in heterogeneous system because it consists of nodes with different speeds, different communication link speeds, different memory sizes, and variable external loads due to the multiple. The numbers of load balancing strategies have been developed and classified so far for getting the high performance of a system

2. Types of ELB

Load Balancing supports three types of load balancers:

- Application Load Balancers
- Network Load Balancers
- Gateway Load Balancer
- Classic Load Balancers

There is a key difference in how the load balancer types are configured. With Application Load Balancers and Network Load Balancers, you register targets in target groups, and route traffic to the target groups. With Classic Load Balancers, you register instances with the load balancer.

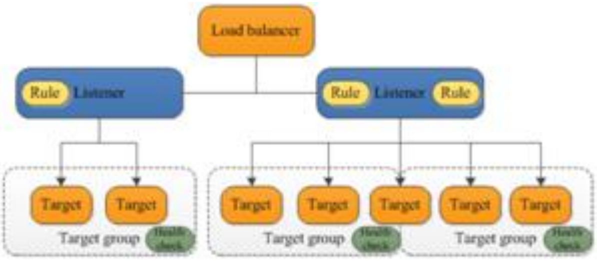


Figure 2: Architecture of AWS Elastic load balancer

2.1 Application Load Balancers

Application load balancer serves as the single point of contact for clients. The load balancer distributes incoming application traffic across multiple targets, such as EC2 instances, in multiple Availability Zones. This increases the availability of your application. You add one or more listeners to your load balancer.

A listener checks for connection requests from clients, using the protocol HTTP and HTTPS and port 80 and 8080 that you configure. Therules that you define for a listener determine how the load balancer routes requests to its registered targets. Each rule consists of a priority, one or more actions, and one or more conditions. When the conditions for a rule are met, then its actions are performed. You must define a default rule for each listener, and you can optionally define additional rules.

Each target group routes requests to one or more registered targets, such as EC2 instances, using the protocol and port number that you specify. You can register a target with multiple target groups. You can configure health checks on a per target group basis. Health checks are performed on all targets registered to a target group that is specified in a listener rule for your load balancer.

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

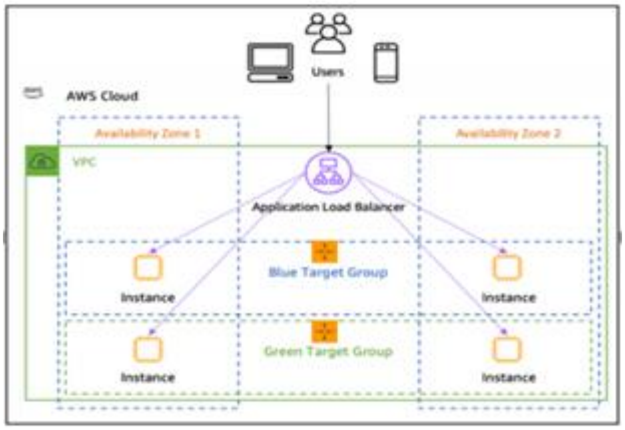


Figure 3: Application load Balancer

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.

2.2 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.

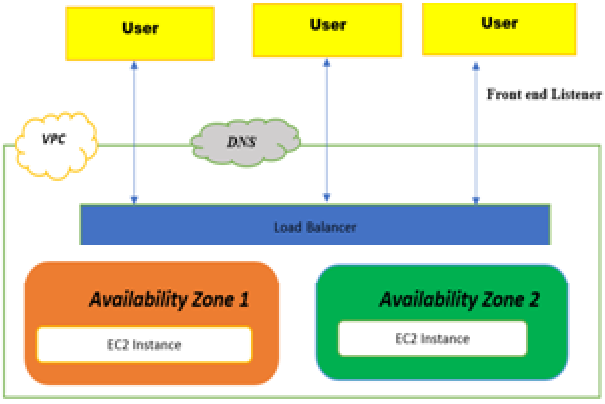


Figure 4: Architecture of Network Load Balancer with AZ

2.3 Gateway Load Balancer

Gateway Load Balancers enable you to deploy, scale, and manage virtual appliances, such as firewalls, intrusion detection and prevention systems, and deep packet inspection systems. It combines a transparent network gateway (that is, a single entry and exit point for all traffic) and distributes traffic while scaling your virtual appliances with the demand.

A Gateway Load Balancer operates at the third layer of the Open Systems Interconnection (OSI) model, the network layer. It listens for all IP packets across all ports and forwards traffic to the target group that's specified in the listener rule. It maintains stickiness of flows to a specific target appliance using 5-tuple (for TCP/UDP flows) or 3-tuple (for non-TCP/UDP flows). The Gateway Load Balancer and its registered virtual appliance instances exchange application traffic using the GENEVE protocol on port 6081. It supports a maximum transmission unit (MTU) size of 8500 bytes.

Gateway Load Balancers use Gateway Load Balancer endpoints to securely exchange traffic across VPC boundaries. A Gateway Load Balancer endpoint is a VPC endpoint that provides private connectivity between virtual appliances in the service provider VPC and application servers in the service consumer VPC. You deploy the Gateway Load Balancer in the same VPC as the virtual appliances. You register the virtual appliances with a target group for the Gateway Load Balancer.

Traffic to and from a Gateway Load Balancer endpoint is configured using route tables. Traffic flows from the service consumer VPC over the Gateway Load Balancer endpoint to the Gateway Load Balancer in the service provider VPC, and then returns to the service consumer VPC. You must create the Gateway Load Balancer endpoint and the application servers in different subnets. This enables you to configure the Gateway Load Balancer endpoint as the next hop in the route table for the application subnet.

2.4 Classic Load Balancer

Classic Load balancer provides basic load balancing across multiple Amazon EC2 instances and operates at both the request level and connection level. our load balancer serves as a single point of contact for clients. This increases the availability of your application. You can add and remove instances 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. As Application Load balancing maintains the load right from the 7th layer that is application layer it provides more efficient way to handle the traffic. The data traffics are handled depending upon the application, hence the paper provides detailed view of simulating the application load balancing.

Security Groups

A security group acts as a virtual firewall that controls the traffic for one or more instances. When you launch an instance, you can specify one or more security groups. You can modify the rules for a security group at any time; the new rules are automatically applied to all instances that are associated with the security group. When we decide whether to allow traffic to reach an instance, we evaluate all the rules from all the security groups that are associated with the instance.

Elastic Load Balancer can be classified two types as per connectivity.

1) Internal Load Balancer
2) Public Load Balancer (Public IP)

An Internet-facing load balancer routes requests from clients over the Internet to targets. An internal load balancer routes requests from clients to targets using private IP addresses.

3. Component of Load balancer

There are three component of load balancer that help to balance incoming traffic between target through target groups.

- Listener (listen the port of traffic)
- Target (Server, Lambda, instance)
- target groups (groups of server and instances)

3.1 Listener

Listener are used to check connection request through port and sent it to requesting port.It is a process that checks for connection requests, using the protocol and port that you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets. default port of ELB eth0
Listeners support the following protocols and ports:
Protocols: HTTP, HTTPS, TCP, UDP, TLS
Ports: 1-65535

Listener rules

Each listener has a default rule, and you can optionally define additional rules. Each rule consists of a priority, one or more actions, and one or more conditions. You can add or edit rules at any time.

3.2 Target Groups

Target groups are group of server or Instance.Target group is used to route requests to one or more registered targets. When you create a listener, you specify a target group for its default action. Traffic is forwarded to the target group that's specified in the listener rule. You can create different target groups for different types of requests.

You define health check settings for your Gateway Load Balancer on a per target group basis. Each target group uses the default health check settings, unless you override them when you create the target group or modify them later on. After you specify a target group in a rule for a listener, the Gateway Load Balancer continually monitors the health of all targets registered with the target group that are in an Availability Zone enabled for the Gateway Load Balancer. The Gateway Load Balancer routes requests to the registered targets that are healthy.

Target group can contain up to 200 Target.

Target can be three types:
1) Instances
2) IP Based (Always within range)
3) Lambda

When target type is IP, we can used only following CIDR.
1) 10.0.0./8
2) 100.64.0.0/10
3) 172.16.0.0/12
4) 192.168.0.0/16

4. Proposed Work Network Load Balancer with Multi AZ

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.

When you enable an Availability Zone for the load balancer, Elastic Load Balancing creates a load balancer node in the Availability Zone. By default, each load balancer node distributes traffic across the registered targets in its Availability Zone only. If you enable cross-zone load balancing, each load balancer node distributes traffic across the registered targets in all enabled Availability Zones.

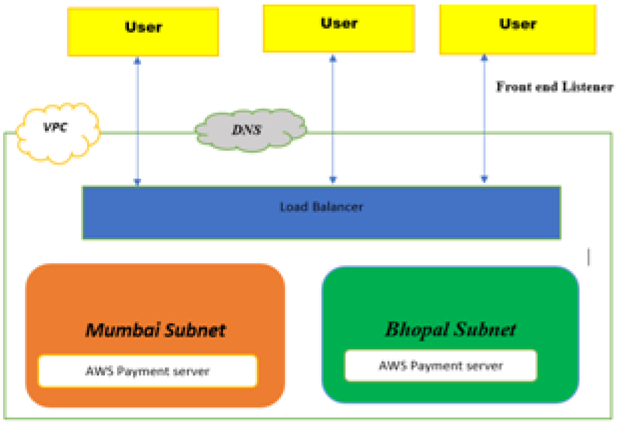


Figure 5: Proposed architecture of Network Load Balancer with Cross zone

If you enable multiple Availability Zones for your load balancer and ensure that each target group has at least one target in each enabled Availability Zone, this increases the fault tolerance of your applications. For example, if one or more target groups does not have a healthy target in an Availability Zone, we remove the IP address for the corresponding subnet from DNS, but the load balancer nodes in the other Availability Zones are still available to route traffic. If a client doesn't honor the time-to-live (TTL) and sends requests to the IP address after it is removed from DNS, the requests fail.

For TCP traffic, the load balancer selects a target using a flow hash algorithm based on the protocol, source IP address, source port, destination IP address, destination port, and TCP sequence number. The TCP connections from a client have different source ports and sequence numbers, and can be routed to different targets. Each individual TCP connection is routed to a single target for the life of the connection For UDP traffic, the load balancer selects a target using a flow hash algorithm based on the protocol, source IP address, source port, destination IP address, and destination port. A UDP flow has the same source and destination, so it is consistently routed to a single target throughout its lifetime. Different UDP flows have different source IP addresses and ports, so they can be routed to different targets.

Elastic Load Balancing creates a network interface for each Availability Zone you enable. Each load balancer node in the Availability Zone uses this network interface to get a static IP address. When you create an Internet-facing load balancer, you can optionally associate one Elastic IP address per subnet.

When you create a target group, you specify its target type, which determines whether you register targets by instance ID or IP address. If you register targets by instance ID, the source IP addresses of the clients are preserved and provided to your applications. If you register targets by IP address, the source IP addresses are the private IP addresses of the load balancer nodes.

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.

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.

Benefits of migrating from a Classic Load Balancer

- Using a Network Load Balancer instead of a Classic Load Balancer has the following benefits:
- Ability to handle volatile workloads and scale to millions of requests per second.
- Support for static IP addresses for the load balancer. You can also assign one Elastic IP address per subnet enabled for the load balancer.
- Support for registering targets by IP address, including targets outside the VPC for the load balancer.
- Support for routing requests to multiple applications on a single EC2 instance. You can register each instance or IP address with the same target group using multiple ports.
- 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 Amazon 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.

Network load balancing is done by creating EC2 instances in 2 separate availability zone and enable cross zone load balancing. AWS cloud provides the service to create the instances and run the simulation tocheck for balancing the requests of user depending upon the incoming load on Network.

Elastic Load balancer should always be accessed using DNS not through IP.

5. Implementation of Server and ELB

Following are the steps followed to implement load balancing:

1) Create two EC2 instance using Microsoft 2019
2) Create Virtual private cloud and subnet
3) Configure a Load Balancer and a Listener
4) Configure Security Settings for an HTTPS Listener
5) Configure a Security Group
6) Configure a Target Group
7) Configure Targets for the Target Group
8) Create the Load Balancer

Considering a small example of two application the implementation is done such that depending upon the load on network the requests are balanced between available target. Two servers are created with EC2 instances, name as AWS payment server Bhopal and AWS payment server Mumbai. each server has attached with load balancer.As shown in Fig.6&7 EC2 instances are created with the names AWS Payment server in Bhopal and Mumbai region and will check load balancer, balancing load of incoming network traffic using TCP protocol with port number 80.

An EC2 instance is a virtual server in Amazon’s Elastic Compute Cloud (EC2) for running applications on the Amazon Web Services (AWS) infrastructure. AWS is a comprehensive, evolving cloud computing platform; EC2 is a service that allows business subscribers to run application programs in the computing environment. The EC2 can serve as a practically unlimited set of virtual
machines.

Amazon provides a variety of types of instances with different configurations of CPU, memory, storage, and networking resources to suit user needs. Each type is also available in two different sizes to address workload requirements.

Instance types are grouped into families based on target application profiles. These groups include: general purpose, compute-optimized, GPU instances, memory optimized, storage optimized and micro instances.

Table 1.0: Configuration Parameter of AWS payment Instance

Parameter	EC2 Mumbai	EC2 Bhopal
Server Name	AWS Payment Mumbai	AWS Payment Bhopal
Instance Type	GPU-T2-Micro	GPU-T2-Micro
VPC Id.	C8e01aa3	C8e01aa3
Subnet	0bb45760	07c2934b
Version	Microsoft window server 2019	Microsoft window server 2019
Availability Zone	ap-south-1a	ap-south-1b
Storage Type/Volume	GPU-SSD-/30GB	GPU-SSD-/60GB
Public IP	13.126.226.203	13.235.49.80
Web server	IIS web page	IIS page
Security Group	Protocol-TCP Port-80,3389,0-65535	Protocol-TCP Port-80,3389,0-65535

In this research work we created two servers inside a virtual private cloud in two different availability zone Mumbai and Bhopal. Depicted in figure 6 and 7 while figure 8 showing utilization report of server.

As shown in Fig.6 EC2 instances are created with the names AWS Payment server.

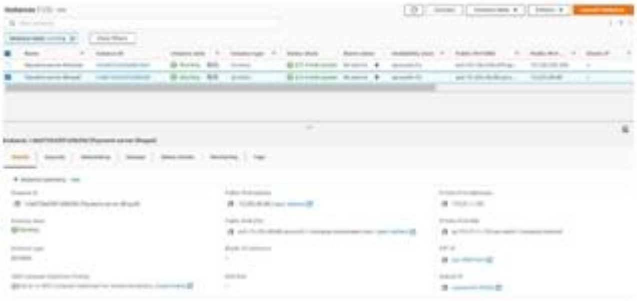


Figure6: AWS payment server Bhopal

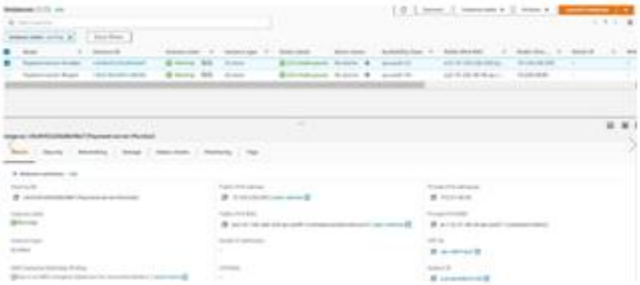


Figure 7: AWS payment server Mumbai

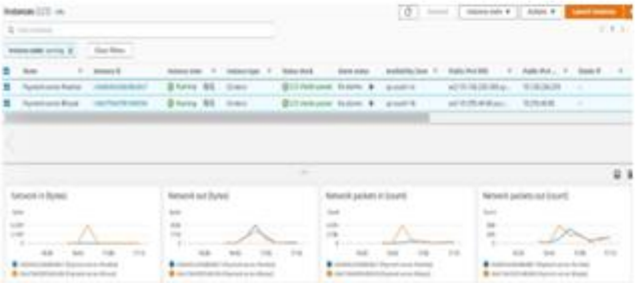


Figure8: CPU and Network utilization graph of server

Amazon Virtual Private Cloud (Amazon VPC) enables you to launch AWS resources into a virtual network that you've defined. This virtual network closely resembles a traditional network that you'd operate in your own data center, with the benefits of using the scalable infrastructure of AWS. In research work we created VPC in India region and two subnets. Figure 9 shows configuration details of VPC.

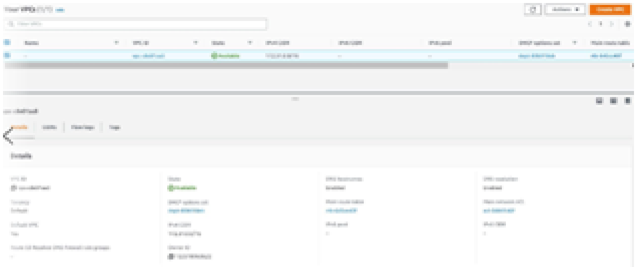
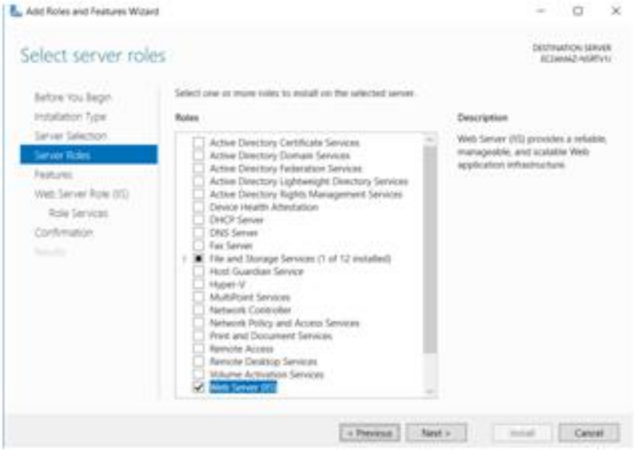


Figure 9: Configuration of Virtual private cloud

When both EC2 instance created then login both server using public IP through remote desktop protocol (RDP) and create IIS web page on both server name as AWS Payment server Bhopal and Mumbai.

Step to create IIS Web page on window server 2019.

1) Login server manager of EC2 Instance.
2) Configure add role and feature and create IIS Web server.
3) Install IIS web server.
4) Create WEB page



Add Roles and Features Wizard

Select server roles

DESTINATION SERVER

Before You Begin
Installation Type
Server Selection
Server Roles
Features
Web Server Role (IIS)
Role Services
Confirmation
Results

Select one or more roles to install on the selected server.

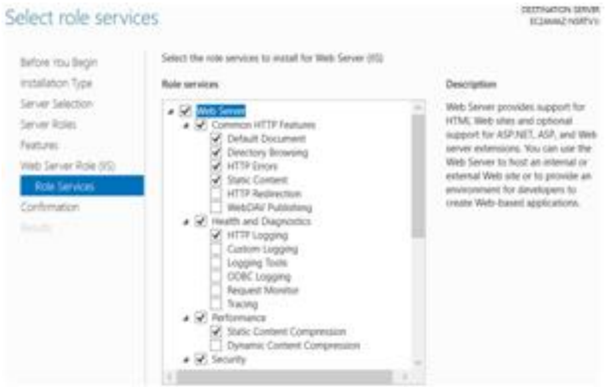
Roles

- ☐ Active Directory Certificate Services
- ☐ Active Directory Domain Services
- ☐ Active Directory Federation Services
- ☐ Active Directory Lightweight Directory Services
- ☐ Active Directory Rights Management Services
- ☐ Device Health Attestation
- ☐ DHCP Server
- ☐ DNS Server
- ☐ Fax Server
- ■ File and Storage Services (1 of 12 installed)
- ☐ Host Guardian Service
- ☐ Hyper-V
- ☐ MultiPoint Services
- ☐ Network Controller
- ☐ Network Policy and Access Services
- ☐ Print and Document Services
- ☐ Remote Access
- ☐ Remote Desktop Services
- ☐ Volume Activation Services
- ☑ Web Server (IIS)

Description

Web Server (IIS) provides a reliable, manageable, and scalable Web application infrastructure.

Figure10: Create IIS web server in Microsoft 2019 serve



Select role services

DESTINATION SERVER

Before You Begin
Installation Type
Server Selection
Server Roles
Features
Web Server Role (IIS)
Role Services
Confirmation
Results

Select the role services to install for Web Server (IIS)

Role services

- ☑ Web Server
  - ☑ Common HTTP Features
    - ☑ Default Document
    - ☑ Directory Browsing
    - ☑ HTTP Errors
    - ☑ Static Content
    - ☐ HTTP Redirection
    - ☐ WebDAV Publishing
  - ☑ Health and Diagnostics
    - ☑ HTTP Logging
    - ☐ Custom Logging
    - ☐ Logging Tools
    - ☐ ODBC Logging
    - ☐ Request Monitor
    - ☐ Tracing
  - ☑ Performance
    - ☑ Static Content Compression
    - ☐ Dynamic Content Compression
  - ☑ Security

Description

Web Server provides support for HTML Web sites and optional support for ASP.NET, ASP, and Web server extensions. You can use the Web Server to host an internal or external Web site or to provide an environment for developers to create Web-based applications.

Figure11: Create IIS web server in Microsoft 2019 server

Security group is a virtual stateful firewall work at ENI level up to 5 security group per EC2 instance interface can be applied. It has only permit rule. figure shows that RDP, HTTP and TCP protocol allow with port number 3389,80 in load balancer security groups it means any traffic which one coming from port number 3389 and 80 will allow otherwise security group deny all other traffic.

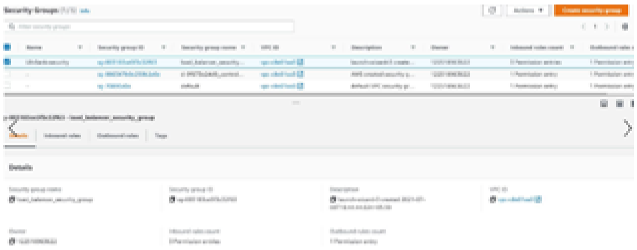


Figure 12: Configuration of security group in AWS console

These statuses are maintained to be 'running'. Once the instances created there has to be some respective targets. Target are instance, server, Lambda or IP address. we used two target which located in two different region Bhopal and Mumbai. Targets are created with respective names as AWS Payments.shown in Fig.9 show the healthy and unhealthy status of server which attached within in target groups. from figure it's clear that both servers are active and working fine.

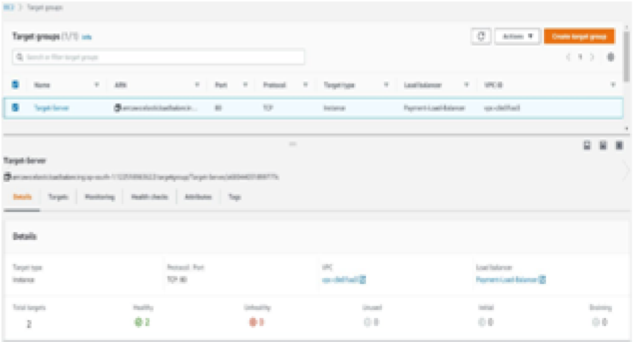


Figure13: Healthy and Unhealthy status of target

Once the target is healthy, the Network load balancer is added to the targets. Type is maintained to be Application and state should be active. This is shown in Fig.13

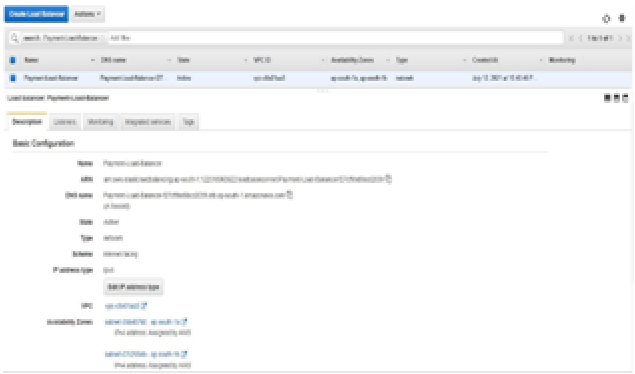
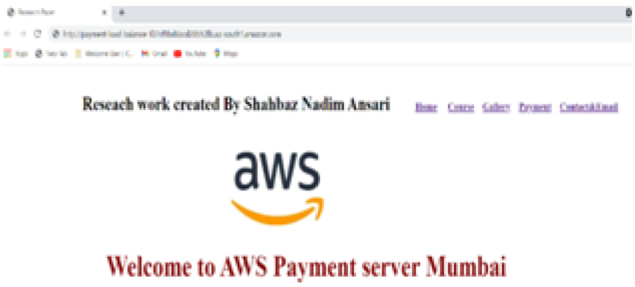


Figure 14: Configuration of Network load balancer and status details

As load balance is ready to balance the traffic but listeners have to be added to the respective targets hence 1st script is written which performs the main objective of balancing depending upon the request made by user. The sample scripts are written to Network balancer to HTTP page with headers as written in the script. Depending upon the requests made the server is changed to the respective instances.

When users accessing AWS DNS Load Balancer http://payment-load-balancer-f27cf9bd6bcd2059.elb.ap-south-1.amazonaws.com/

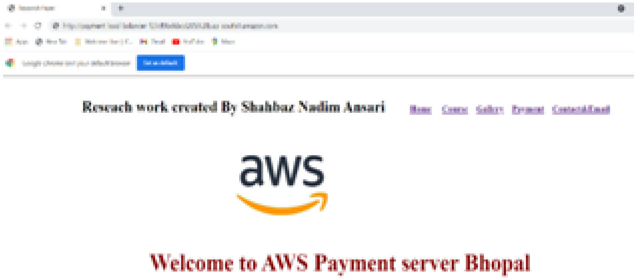
Then all incoming traffic from internet side, Network load balancer balancing incoming traffic and equally distributed between available target toward Mumbai and Bhopal server.



Reseach work created By Shahbaz Nadim Ansari

Welcome to AWS Payment server Mumbai

Figure 15: User Login AWS payment server from Mumbai server



Reseach work created By Shahbaz Nadim Ansari

Welcome to AWS Payment server Bhopal

Figure 16: User Login AWS payment server from Bhopal server

Figure 15 and 16 illustrate that when user accessing AWS payment site then Network load balancer distribute incoming traffic between two AWS payment server located inMumbai and Bhopal regionwhich installed in two separate Availability zone. hence aim of this paper is to designed and implementation of Network load balancer in Multi AZ successfully achieved.

6. Conclusion

Load balancing plays a very important role as usage of internet is growing day by day. As usage increased the traffic in the network increases hence Network load balancer provides the features which can help to resolve the problems and provide easy solution to redirect the paths of request to the servers based on the request made. Instead of dividing the path when based on freeness of servers it is made based on the Network.

7. Future Work

After successfully designed network load balancer our aim would be to design Application and Network Load balance with autoscaling. AWS provide auto scaling feature where server automatically created with the configuration of existing parameter. when load exceeded from limited range then autoscaling feature of AWS create server automatic with scale in and scale out feature. This work will help to overcome the problem of server down.

References

[1] “A Comparative Study of Static and Dynamic Load Balancing Algorithms in Cloud Computing”, International Conference on Energy, Communication, Data Analytics and Soft Computing, 2017

[2] Mari Marios D. Dikaiakos, George Pallis, DimitriosKatsaros, Pankaj Mehra, Athena Vakali (2009, Oct.). Cloud Computing: Distributed Internet Computing for IT and Scientific Research. IEEE Internet Computing, vol. 13(issue 5), pp. 10-13.

[3] Panagiotis Kalagiakos, Panagiotis Karampelas, "Cloud Computing Learning" in the Proceeding of IEEE International Conference on Application of Informat.

[4] Shridhar G.Domanal and G. Ram Mohana Reddy, “Optimal Load Balancing in Cloud Computing By Efficient Utilization of Virtual Machines” in the Proceeding of the IEEE International Conference on Communication Systems and Networks, Bangalore, Jan. 2014, pp. 1-4.

[5] “Static load balancing technique for geographically partitioned public cloud”, Scalable Computing: Practice and Experience, 2019

[6] “Load balancing algorithm in cloud computing”, 5th IEEE International Conference on Parallel, Distributed and Grid Computing (PDGC-2018), 20-22 Dec, 2018

[7] Jitendra Bhatia, MalaramKumhar, "Perspective Study on Load Balancing Paradigms in Cloud Computing," IJCSC Vol- 6 • Issue-1 Sep - Mar 2015 pp.112-120