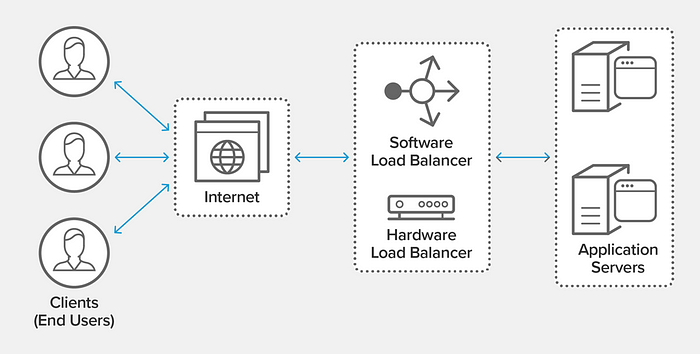
**LOAD BALANCERS**

Load Balancing is the practice of distributing network traffic or computational workloads between two or more computers. Most modern applications have to process millions of request from it’s users simultaneously. A load balancer acts as a traffic controller which sits in front of your servers as a facilitator ensuring that all servers are used equally. A load balancer might be a physical device, a software process or a virtualized instance running on a specialized hardware.

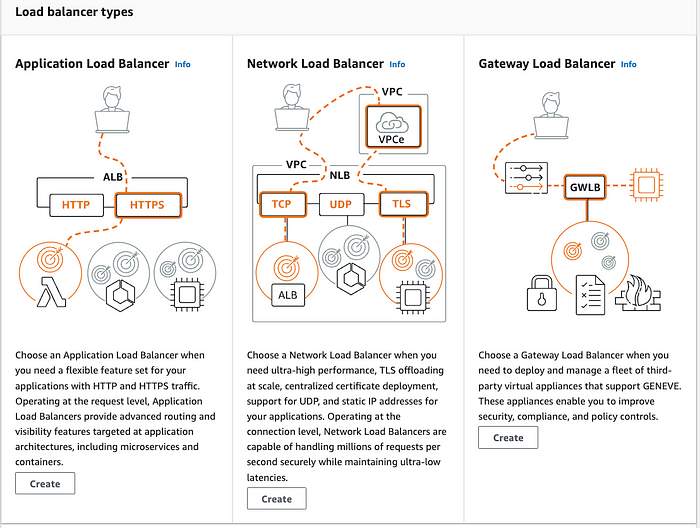
**How load balancers work.**



what is load balancing

* When a request arrives the load balancer from a user, the load balancer sends the request to a given server. The load balancer repeats that process for all incoming request.
* Consider a restaurant scenario where a restaurant has fewer customers that the number of waiters. I customers are allowed to select the waiters they want then some waiters would be idle while others would be overloaded with jobs. Let’s say we employ a manager (load balancer) who is responsible for assigning customers to waiters. The manager now ensures no waiter would be idle and also make sure no waiter would be overloaded.
* Load balancers also detect the health of backend resources and do not send traffic to servers that are not able to fulfill request. Load balancers helps to improve the overall performance and reliability of cloud-based applications by ensuring that resources are used efficiently and that there is no single point of failure
* If there is an increase in network traffic, a load balancer can bring up additional servers online to keep up with the demand during the spike. Also, if there is a fall in network traffic, the load balancer can reduce the amount of servers online.

# What are the 3 Types of Load Balancers in AWS



[Load balancing](https://www.geeksforgeeks.org/what-is-load-balancer-system-design/) is a crucial aspect of ensuring high availability, scalability, and fault tolerance in cloud computing environments. Amazon Web Services (AWS) provides several types of load balancers to distribute incoming traffic across multiple targets, such as instances, containers, and IP addresses.

The three main types of load balancers in AWS are

1. the Classic Load Balancer (CLB),
2. the Application Load Balancer (ALB), and
3. the Network Load Balancer (NLB).

**1. Classic Load Balancer (CLB)**

The Classic Load Balancer is the oldest of the three types and is designed for applications that were built within the EC2-Classic network. It operates at both the application and network layers, distributing traffic across instances based on either application or network-level information.

*While it provides basic load balancing capabilities, CLB lacks some of the advanced features available in the newer load balancers.*

**2. Application Load Balancer (ALB)**

The Application Load Balancer is a Layer 7 load balancer that is ideal for routing HTTP/HTTPS traffic. It is highly suited for modern, microservices-based architectures. ALB can route requests based on content, enabling more advanced load-balancing scenarios.

* It supports host-based and path-based routing, allowing for efficient handling of multiple applications running on the same set of instances.
* Additionally, ALB provides features such as SSL termination, WebSocket support, and native support for containerized applications.

**3. Network Load Balancer (NLB)**

The Network Load Balancer operates at the transport layer (Layer 4) and is designed to handle high-throughput traffic. NLB is well-suited for scenarios where extreme performance and low latency are critical, such as gaming applications or large-scale media streaming.

* It distributes incoming TCP/UDP traffic across a group of targets based on IP protocol data.
* NLB is highly scalable and can handle millions of requests per second, making it suitable for demanding workloads.

**Load balancing algorithms**

A load balancer determines which server to send the request to based on different types of algorithms. Load balancing algorithms provide different capabilities and benefits to satisfy different use cases.

1. **Round robin:**Round robin is a simple load balancing solution for making sure that a virtual server forwards each client request to a different server based on a rotating list. This is the most basic load balancing method. This method uses only the server name or id to determine which server will receive and process the next incoming request.
2. **Weighted Round Robin:**This method creates a queue for incoming request. Here, each server is assigned a “weight”. The weight is used to determine which server should have priority over others to handle an incoming request. This method is best suited to an environment with servers that have different resources. This is because the most powerful server will have the highest weight.
3. **IP hash:**Here, the server performs a mathematical computation called hashing on the client IP address into a smaller value called a *hash key.*The hash key represents the user’s IP address an is used as the basis on how to decide to which server to route the request to.
4. **Least Connections:**Here, the algorithm gives priority to the server with the fewest active connections when a new request is received. This prevents severs from being overloaded with connections. This algorithm is best suited for environments with identical server resources.
5. **Weighted Least Connections:**Unlike the Least Connections algorithm, this algorithm takes into account server resources. Here, the load balancer assigns different weights to each server. As with Weighted Round Robin, the most powerful server has a higher weight. The load balancer the assigns each new request to the server with the lowest active connection-weight ratio.
6. **Least response time:**This algorithm gets the response time that each server takes to process an incoming request and sends a response. This algorithm sends an incoming request to the fastest server with the fewest active connection and lowest response time.

**Benefits of load balancing**

1. **Improved Performance:**Load balancers improve application performance by reducing network latency. It helps to distribute the workload across multiple resources, which reduces the load on each resource and improves the overall performance of the system.
2. **High Availability:**Server failure or maintenance can increase application downtime. A load balancer ensures that there is no single point of failure in the system. This provides higher availability and fault tolerance. If a server is about to fail, or is offline for maintenance or upgrades, load balancing automatically reroutes the workload to a working server to avoid service interruptions and maintain high availability.
3. **Application security:**Load balancers have features that add additional layers of security to your applications. Features such as SSL encryption, web application firewalls (WAF) and multi-factor authentication (MFA) can be used to defend against security risk such as distributed denial-of-service (DDoS) attacks. Also load balancers can monitor traffic and block malicious content.
4. **Scalability:**Load balancers are used to direct network traffic amongst multiple servers. Load balancers helps to handle multiple request concurrently. This prevents overloading a single server. Also, servers can be added or removed as needed making scaling an automated process.

**Types of Load Balancers**

Load balances are of several types based on it’s specifications. Below are the main types of load balancers.

**Global server load balancers**

The Global server load balancer is a type of load balancer used to route traffic across multiple geographical locations. Here, user request are assigned to the closest server. They might redirect traffic to servers outside the client’s geographic zone only in case of server failure.

**Application load balancing**

This load balancers are used for application content such as URLs, SSL sessions and HTTP headers to route API (Application Programming Interface) request. They look at the request content, such as HTTP headers or SSL session IDs, to redirect traffic.

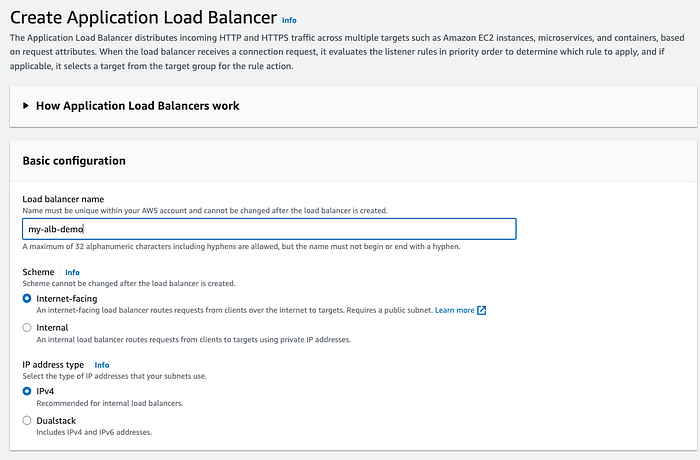
**Network load balancing**

Network load balancers examine IP addresses and other network information to redirect traffic optimally. Network load balancers optimize an reduce latency across local and wide area networks.

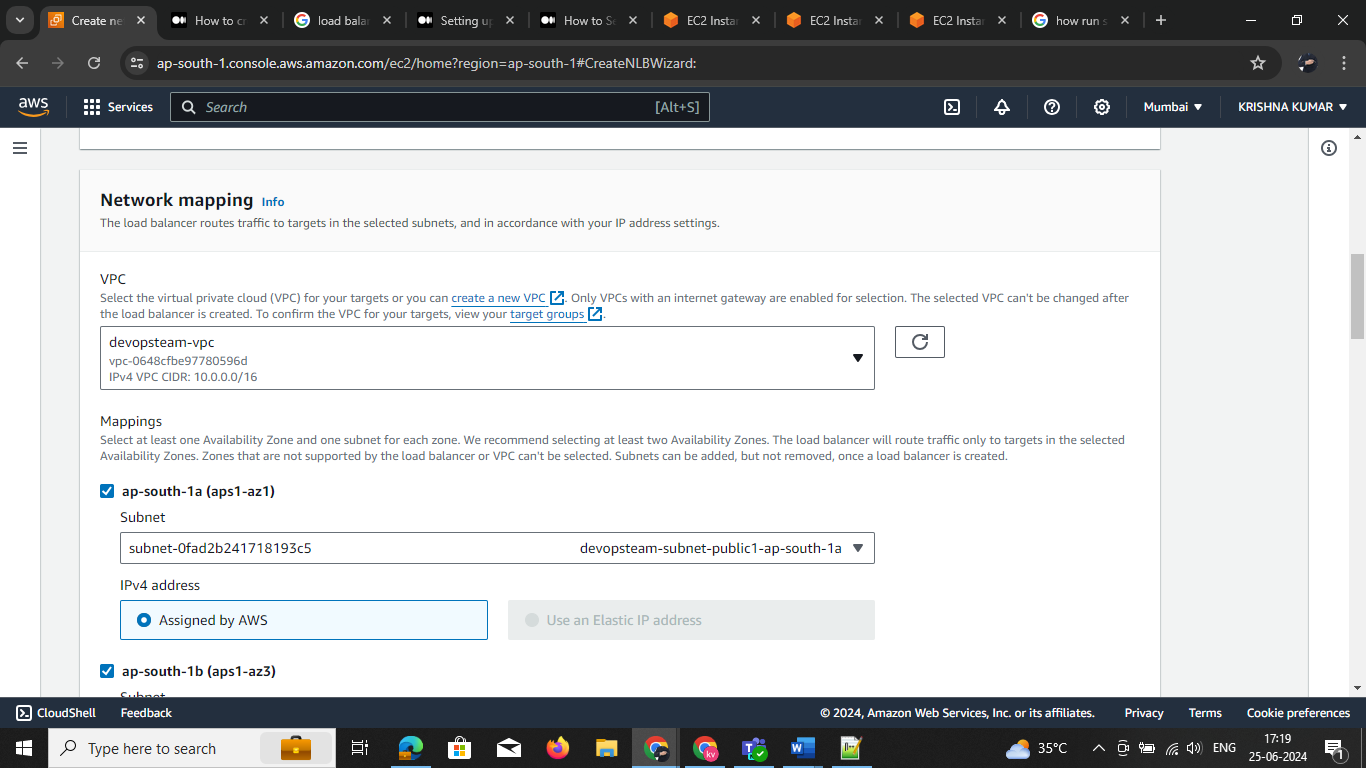
**Application load balancing**

*Give your Load balancer some meaningful name*

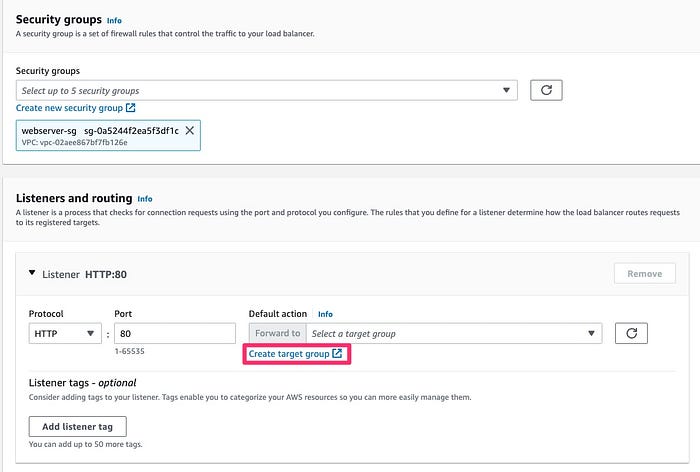
1. *You need to select Scheme(it can be Internet-facing(An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet) or Internal(An internal load balancer routes requests from clients to targets using private IP addresses))*
2. *IP address type(Can be IPv4 or Dualstack(Includes IPv4 and IPv6 addresses))*



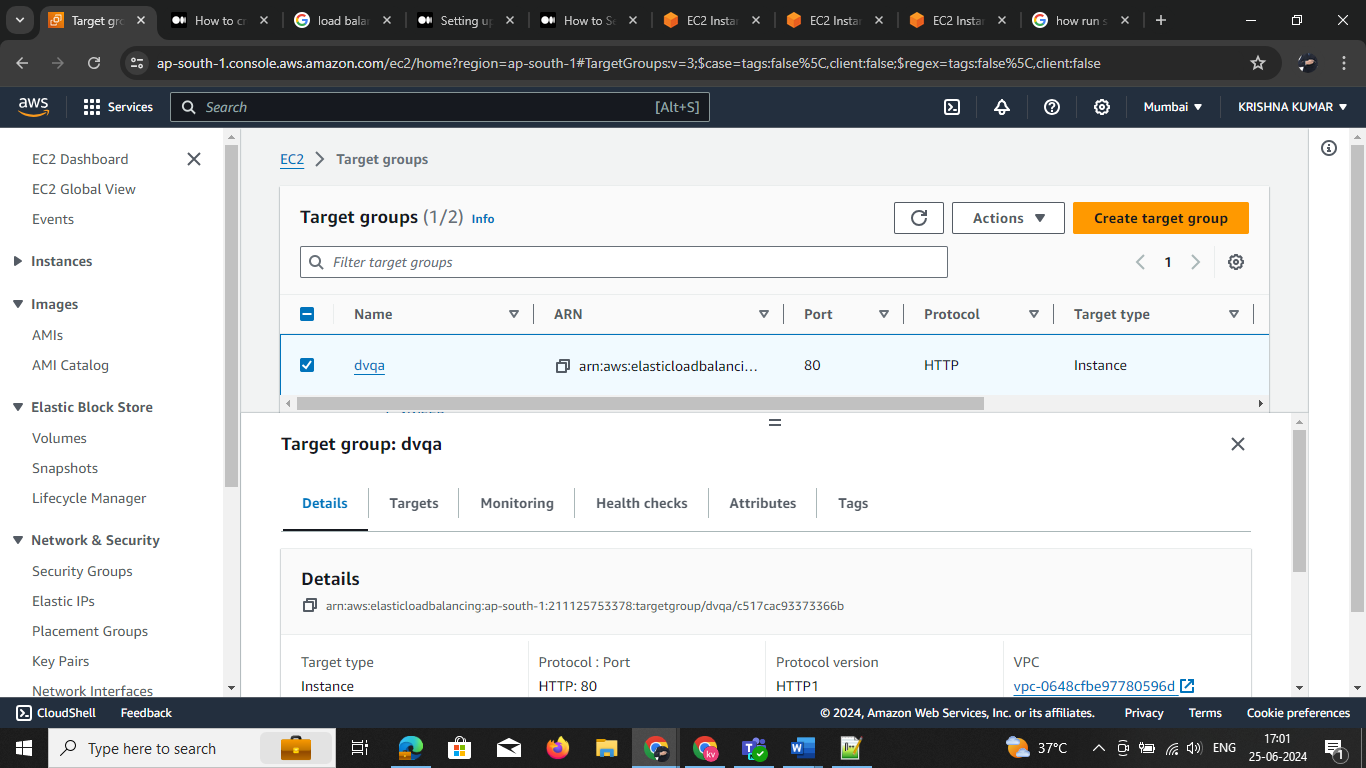
* *Select the VPC and availability zone*

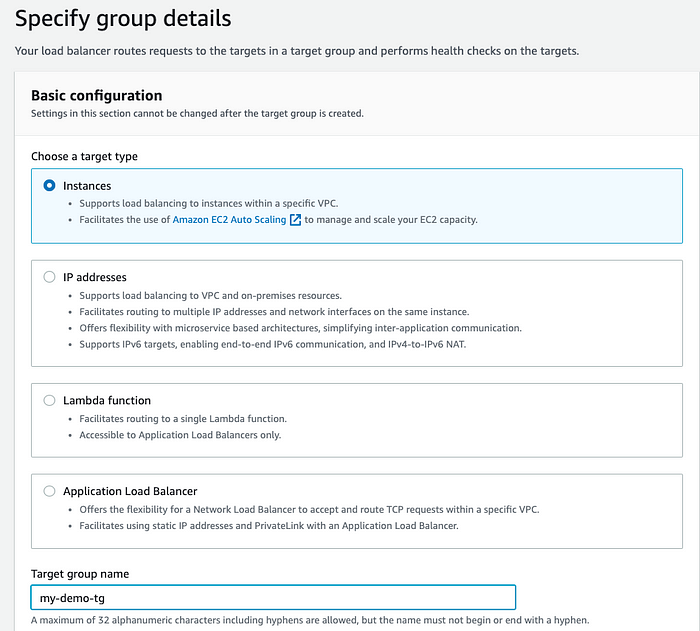


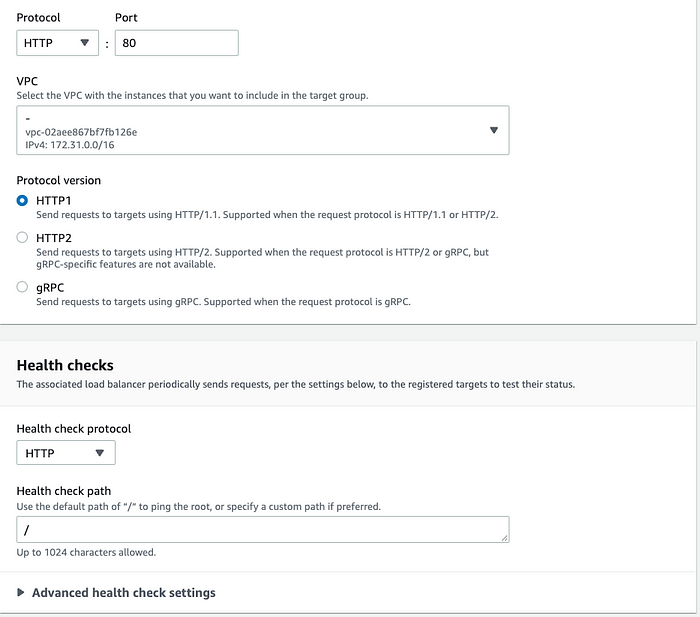
* *Under Listeners and routing, define the port and protocol on which the load balancer must listen. Each Application*
* *Load Balancer needs at least one listener to accept traffic.*



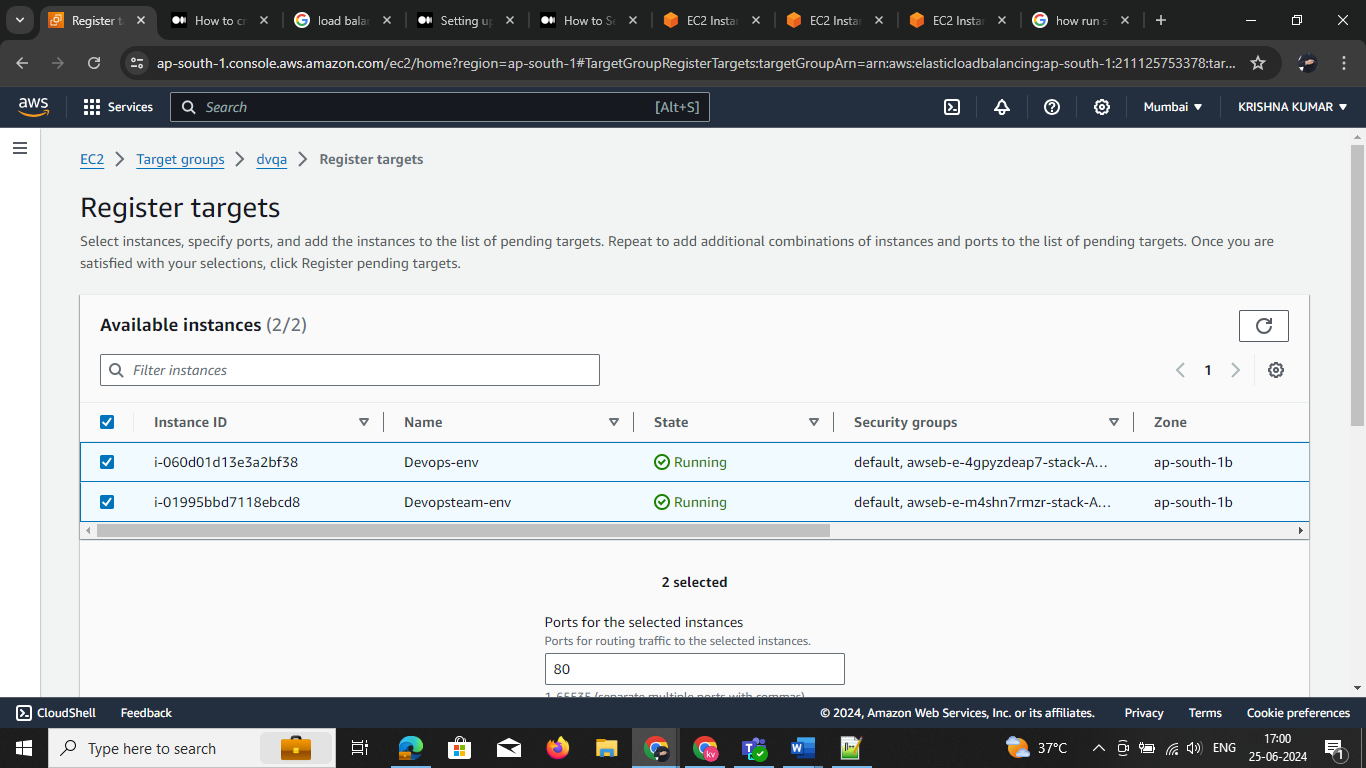
* *Click on the Target group. Target group defines the logical grouping of the targets.*





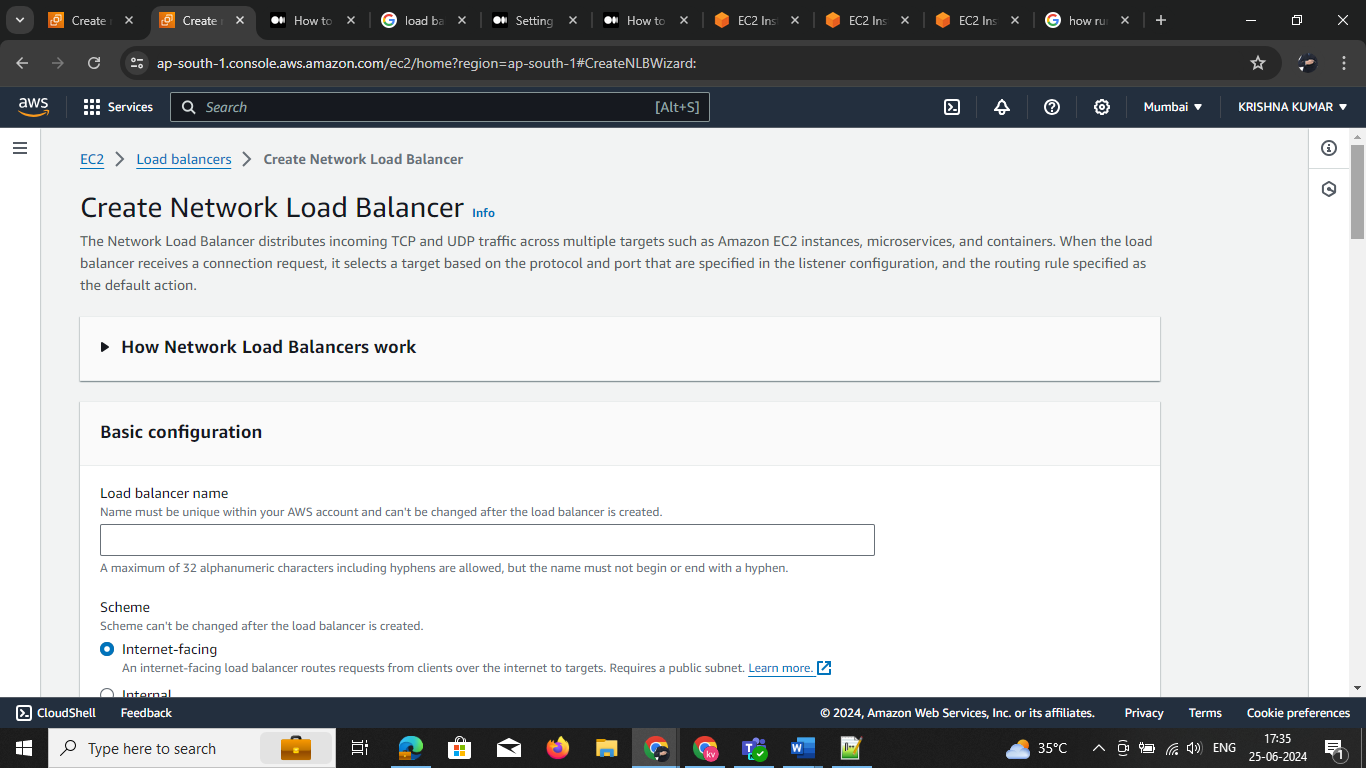


* *Go back to the ALB console and select the target group you have just created. Click on Create load balancer at the bottom of the screen.*

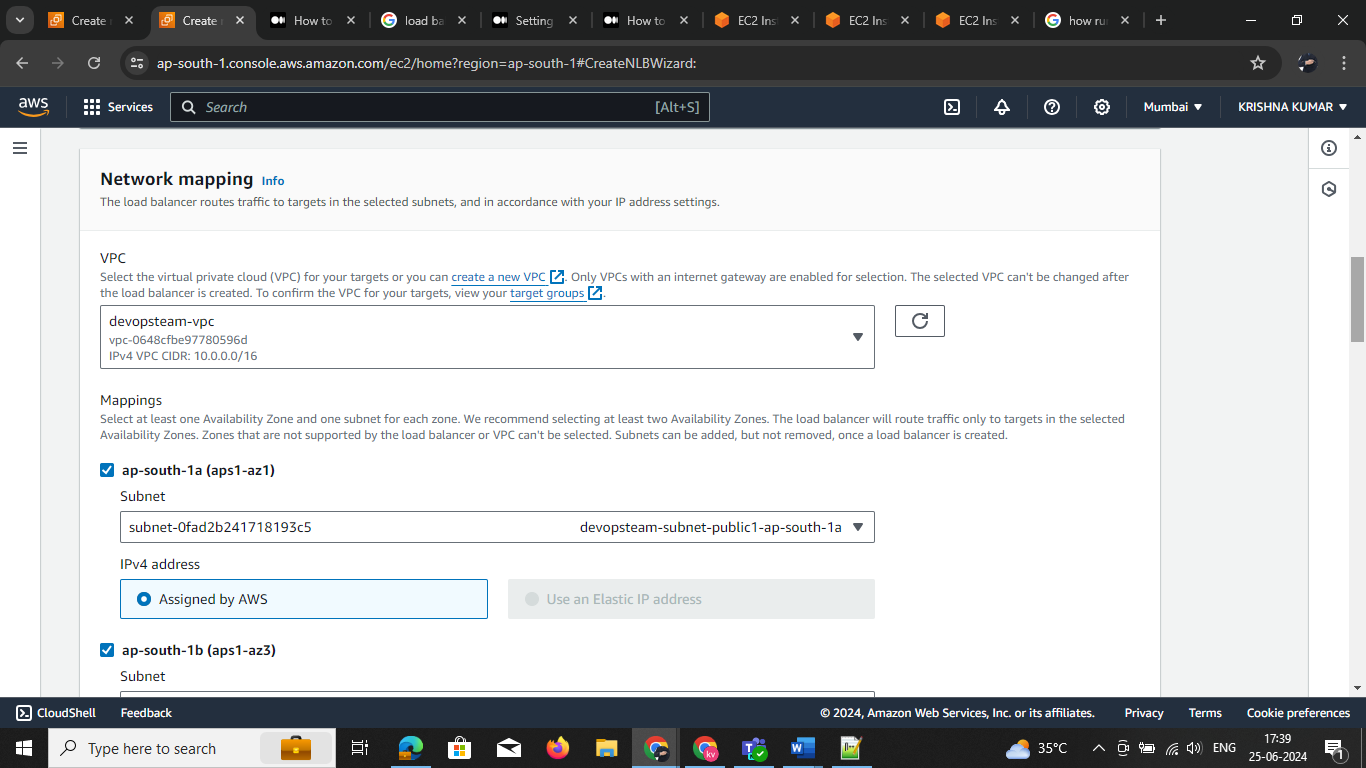


**Network Load Balancer (NLB)**

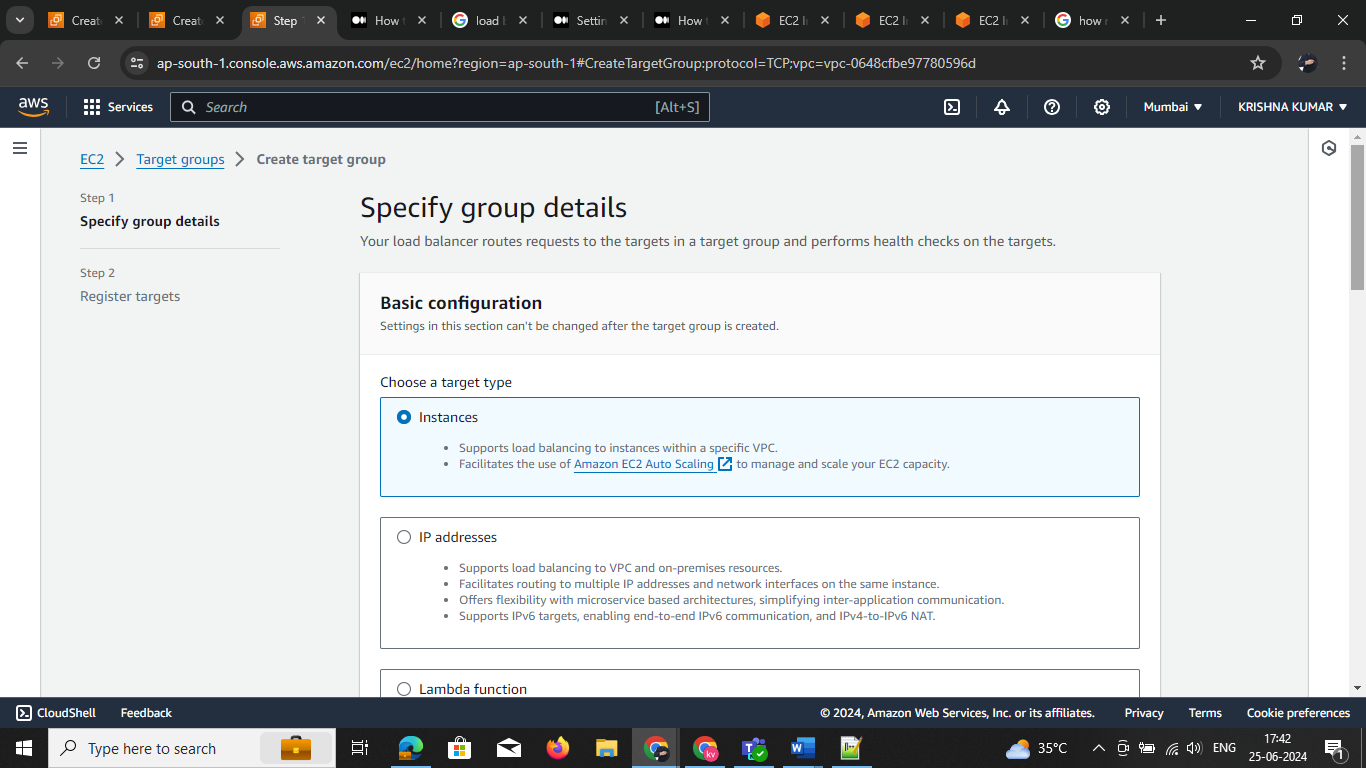
1 Create 🡪 go to ALB -> select the load -🡪 create



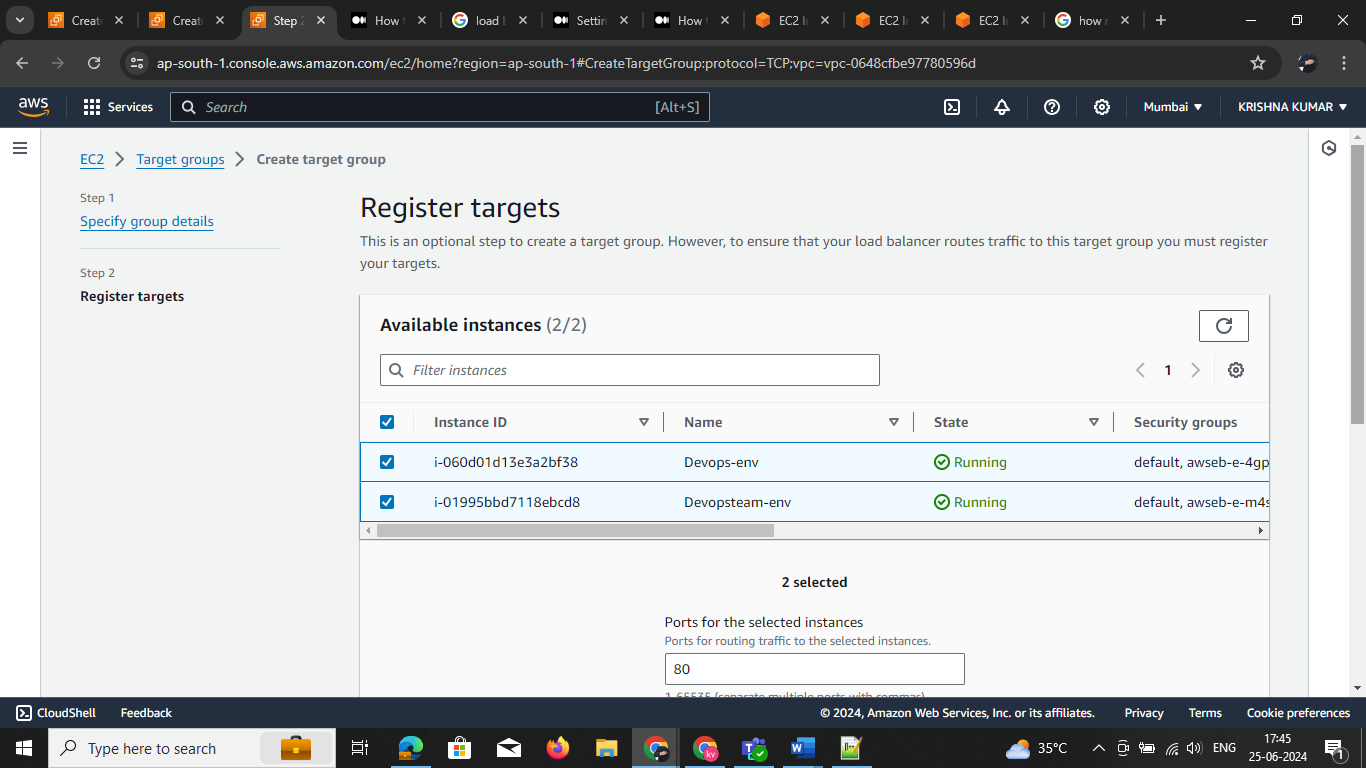
2 Select VPC

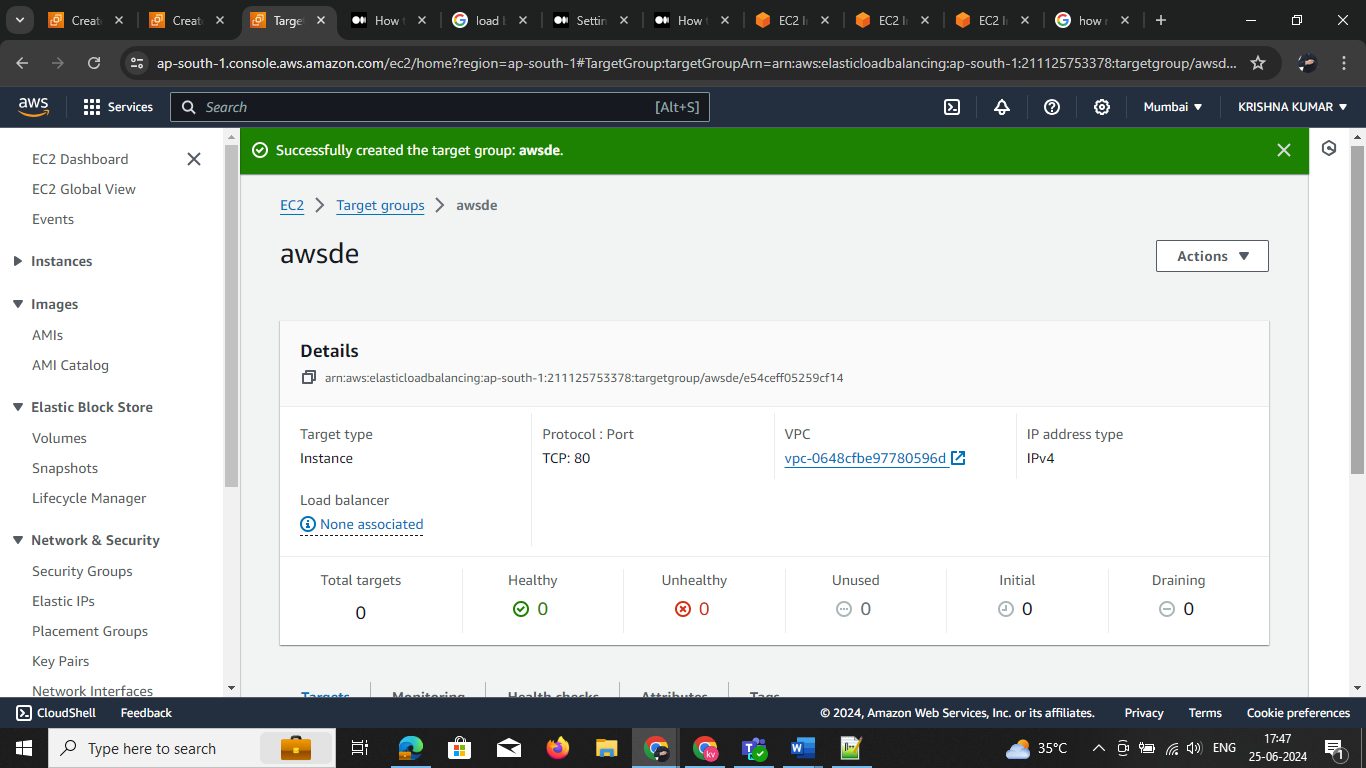


3: creating target group go ec2 -> services or search for target group GIVE Name



4 Register target group and create it

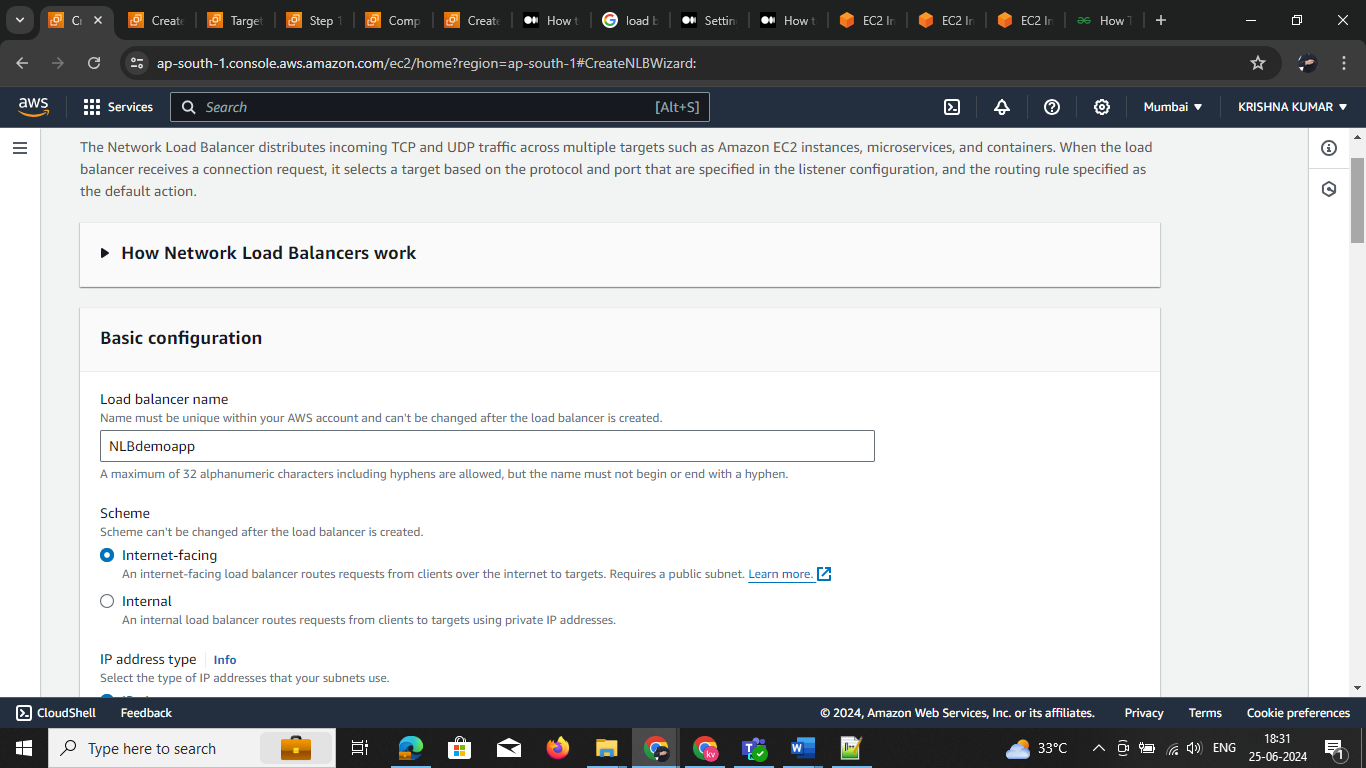




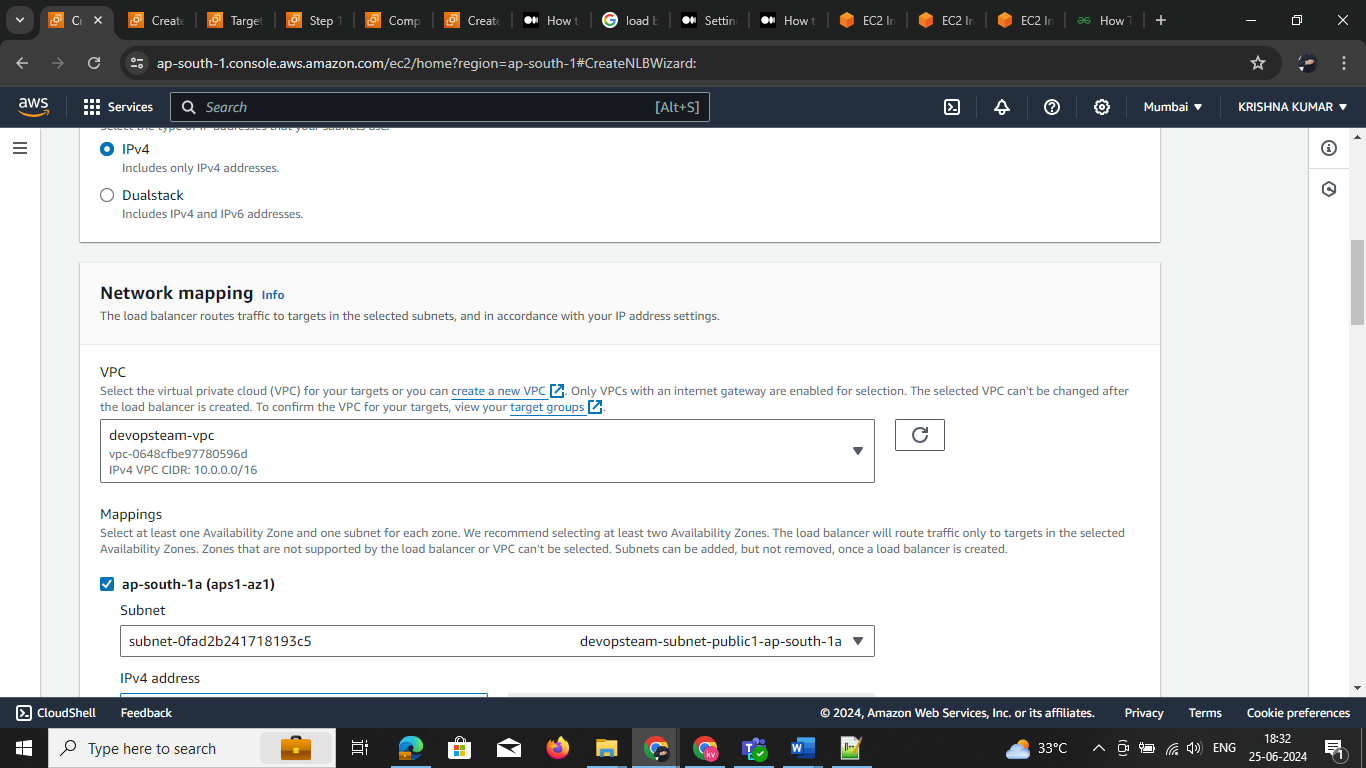
Target group is created

NLB

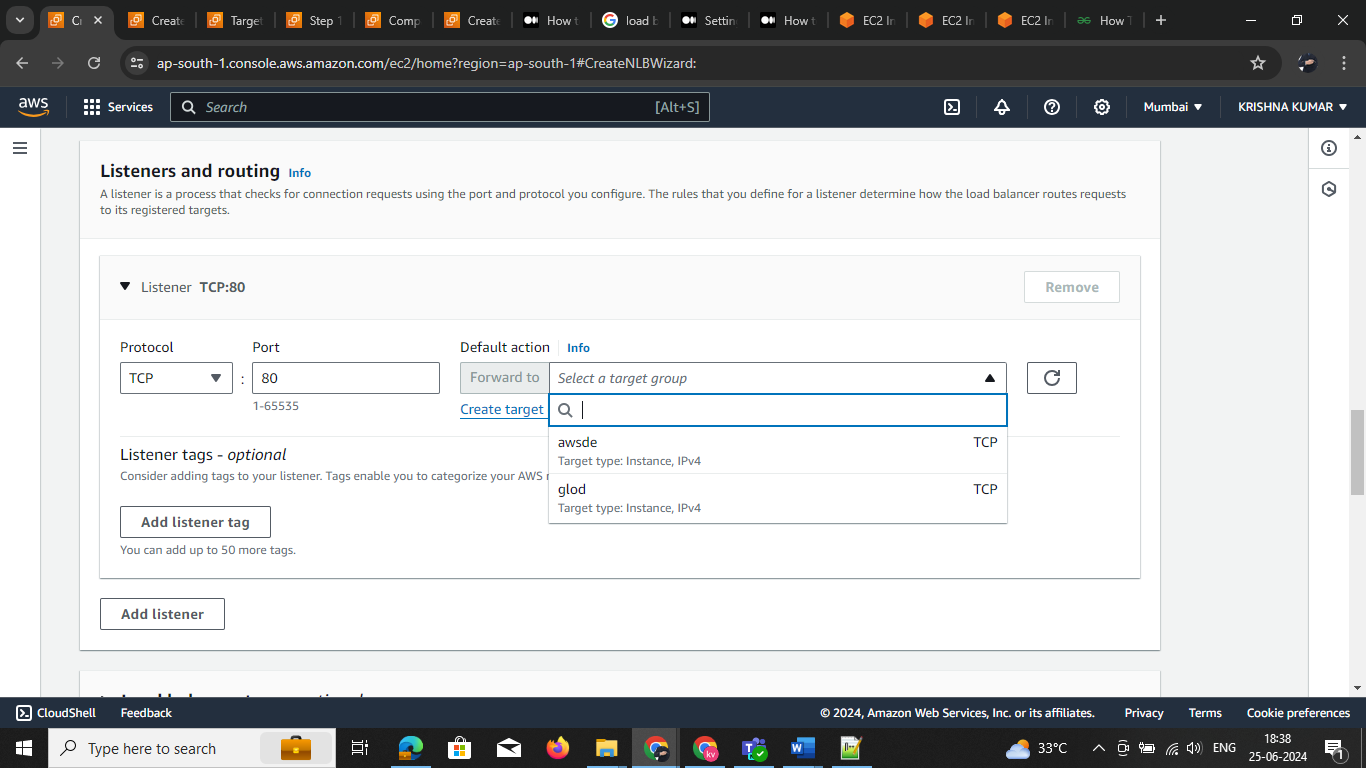
* 1. go 🡪 load balancer -🡪 select NLB click on create



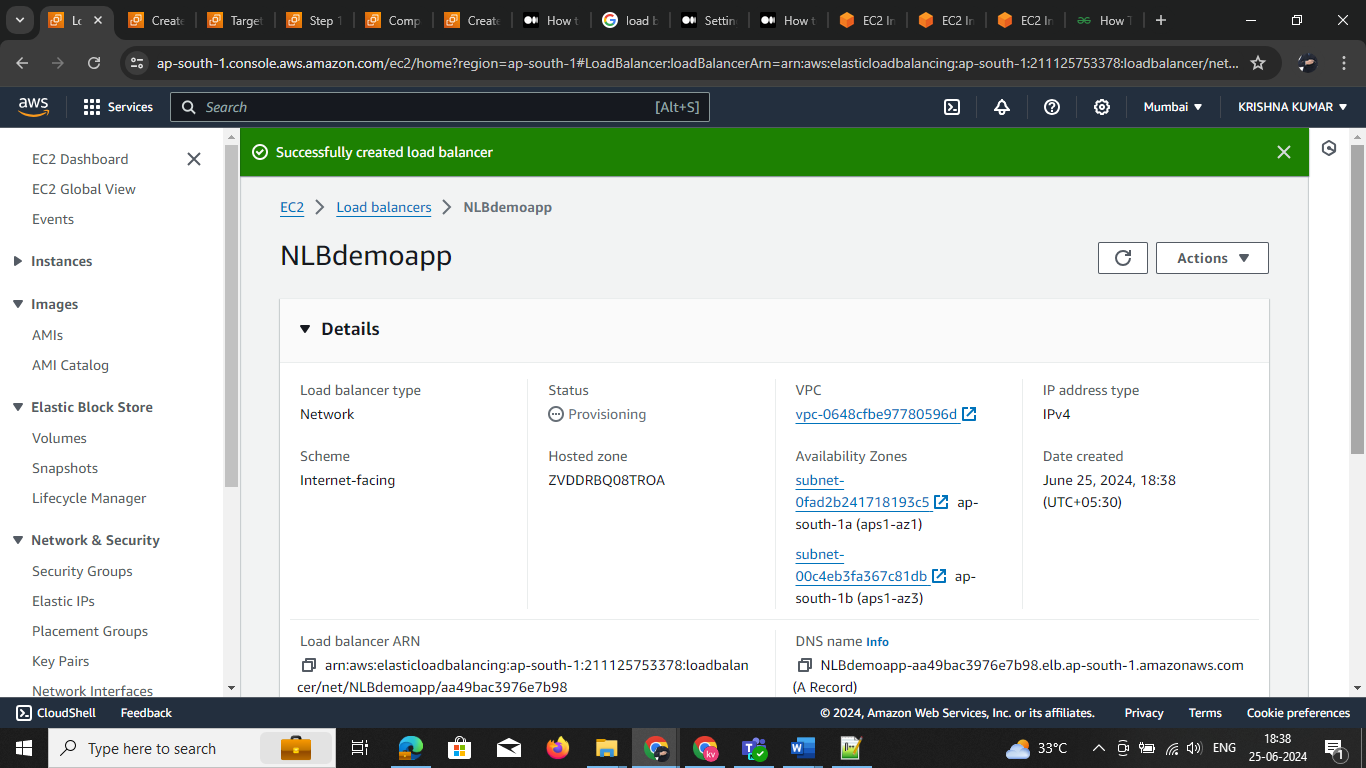
2 Select the vpc and are your custom vpc name select security group exiting are default



3 Select the target group and are create the target group



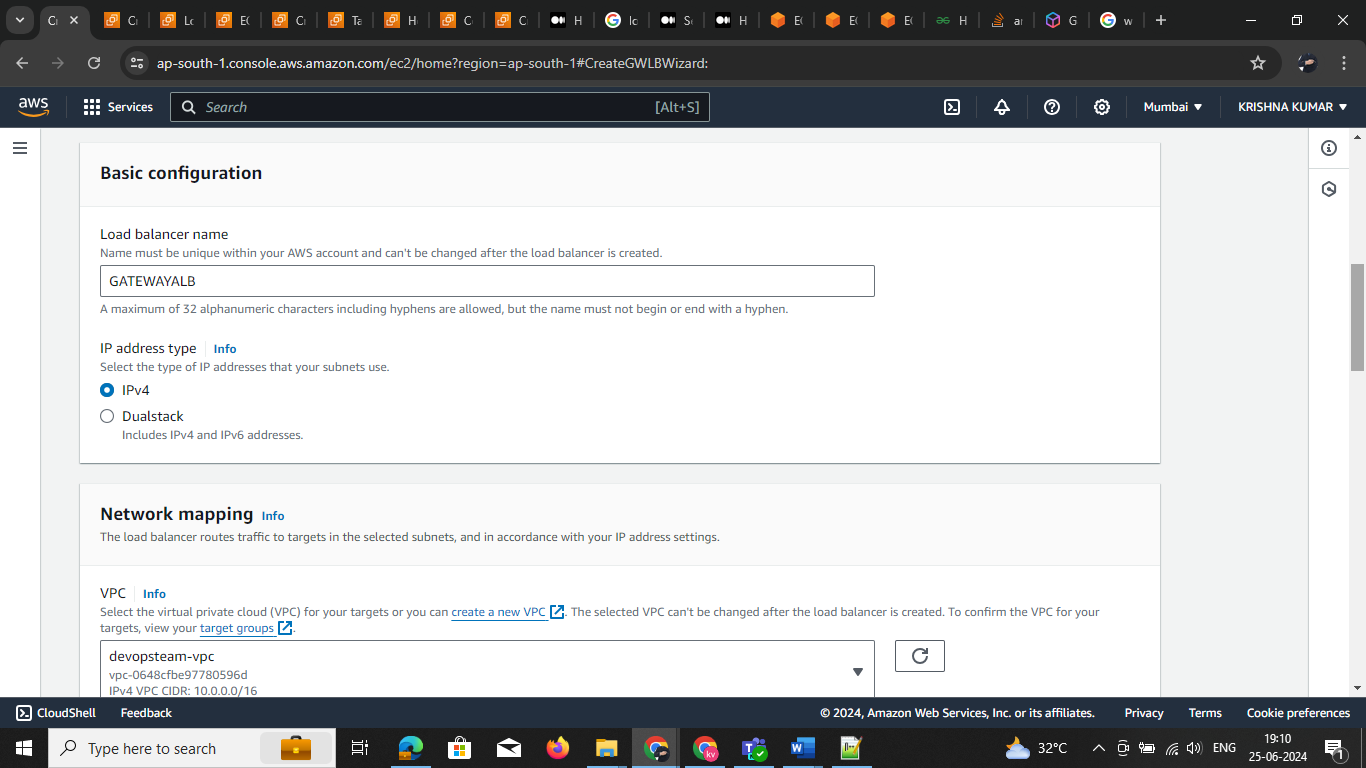
4 Now load balancer is create



**Create Gateway Load Balancer**

When you create a Gateway Load Balancer, you can deploy, scale, and manage third-party virtual appliances that support GENEVE ([RFC 8926](https://tools.ietf.org/html/rfc8926)) such as firewalls and other security systems in the cloud. Gateway Load Balancers use Gateway Load Balancer endpoints to securely exchange traffic across VPC boundaries. Traffic to and from a Gateway Load Balancer endpoint is configured using route tables.

1 go 🡪 load balancer -🡪 select GLB click on create

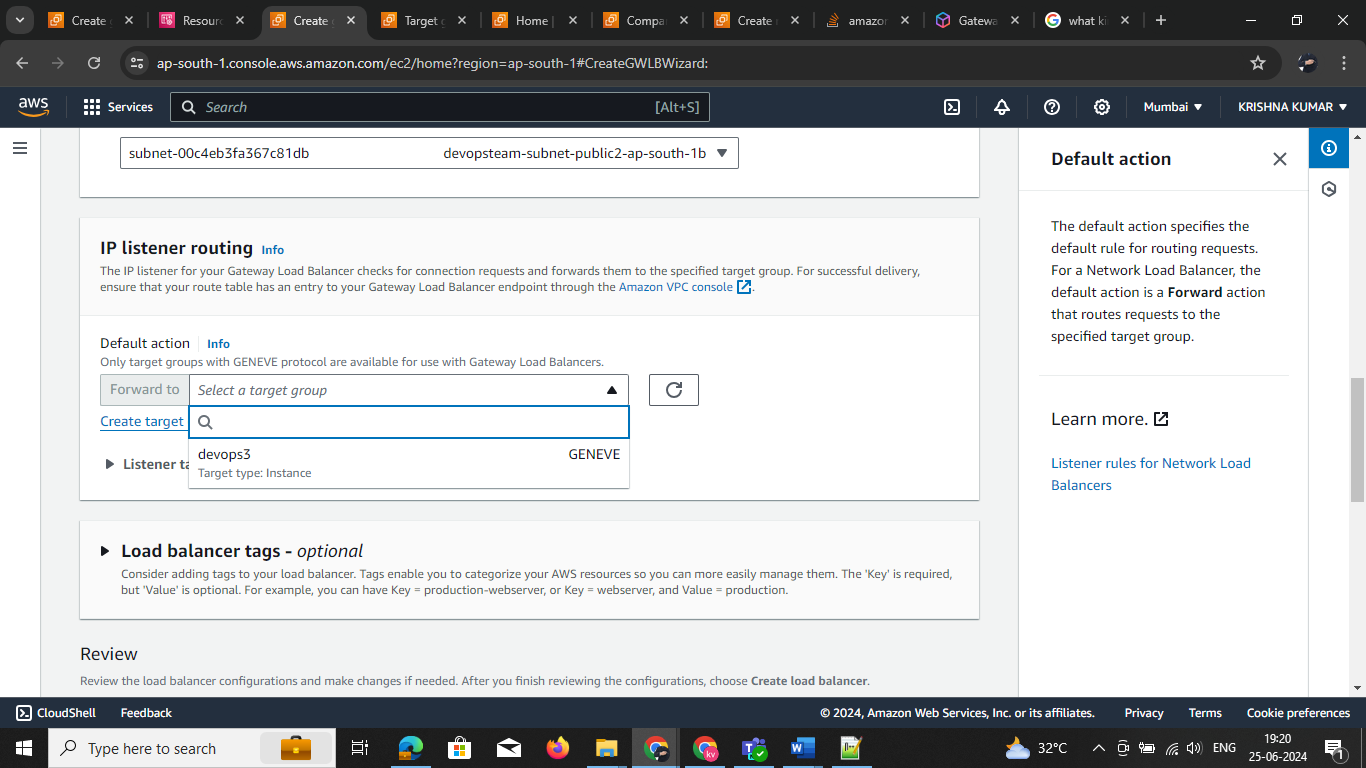


Select the vpc and are your custom vpc name select security group exiting are default

2: **creating target group go to ec2 -> services or search for target group GIVE Name**

THIS load balancer support GENEVE action

Amazon Web Services' (AWS) Gateway Load Balancer (GWLB) supports the GENEVE protocol on port 6081 to exchange application traffic with registered virtual appliance instances. GWLB operates at the third layer of the Open Systems Interconnection (OSI) model, listening for all IP packets across all ports and forwarding traffic to the target group specified in the listener rule



* Now load balancer is create

