**Network Load Balancer**

A **Network Load Balancer (NLB)** in AWS is a highly scalable, high-performance Layer 4 (Transport Layer) load balancer. It's designed to handle **millions of requests per second** while maintaining ultra-low latencies.

**Features of Network Load Balancer (NLB):**

* Network Load Balancer operates at the **4th layer (Transport Layer)** of the Open Systems Interconnection (OSI) model.
* It distributes network traffic based on IP protocol data to targets such as Amazon EC2 instances, micro services, and containers within an Amazon Virtual Private Cloud (VPC).
* It supports load balancing for both TCP and UDP traffic.
* Provides a static IP address per Availability Zone.
* It allows you to assign one Elastic IP address per subnet.
* Supports client TLS (Transport Layer Security) session termination. (TLS is a cryptographic protocol designed to provide communications security over a computer network).
* Preserves (saves) the client's source IP address, allowing backend applications to see the original IP of the client.
* **Health checks** at the target group level.

Let’s work with Network load balance in practically:

Create a Network load balancer and add the target group (Backed pool) with two instances (in which Nginx is installed). Now by using the public IP of Network load balancer brows in any browser.

Step1: Create two EC2 instances in same VPC.

To create the EC2 instance first we have to create and configure the VPC, Subnet, Internet gateway, Rout Table, and security group.

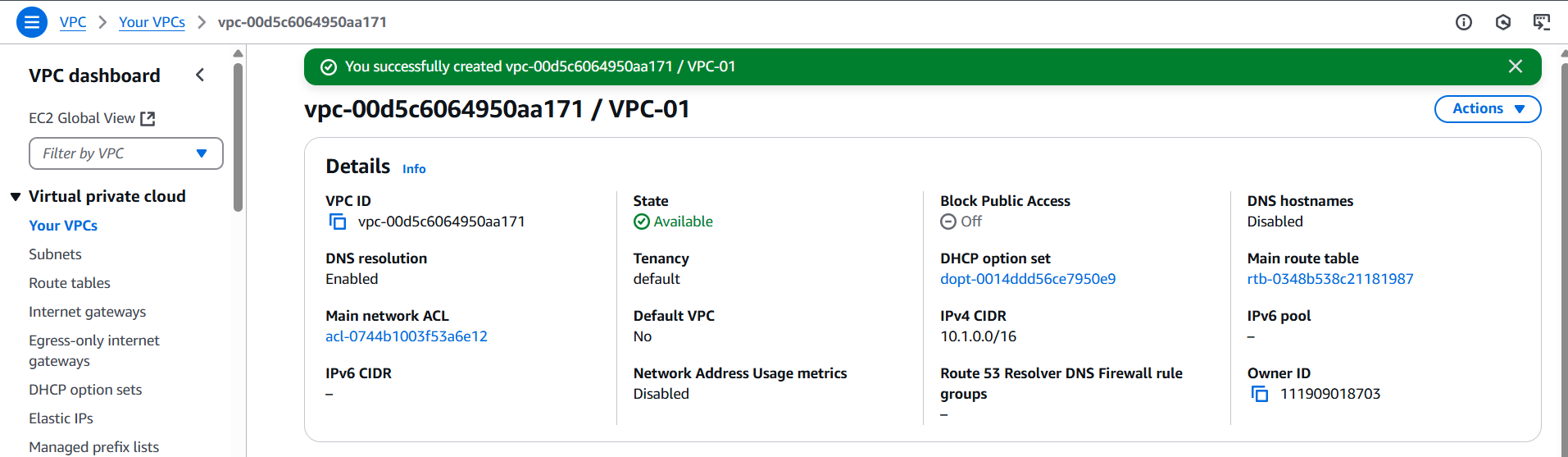


Fig: Virtual private cloud (VPC-01).

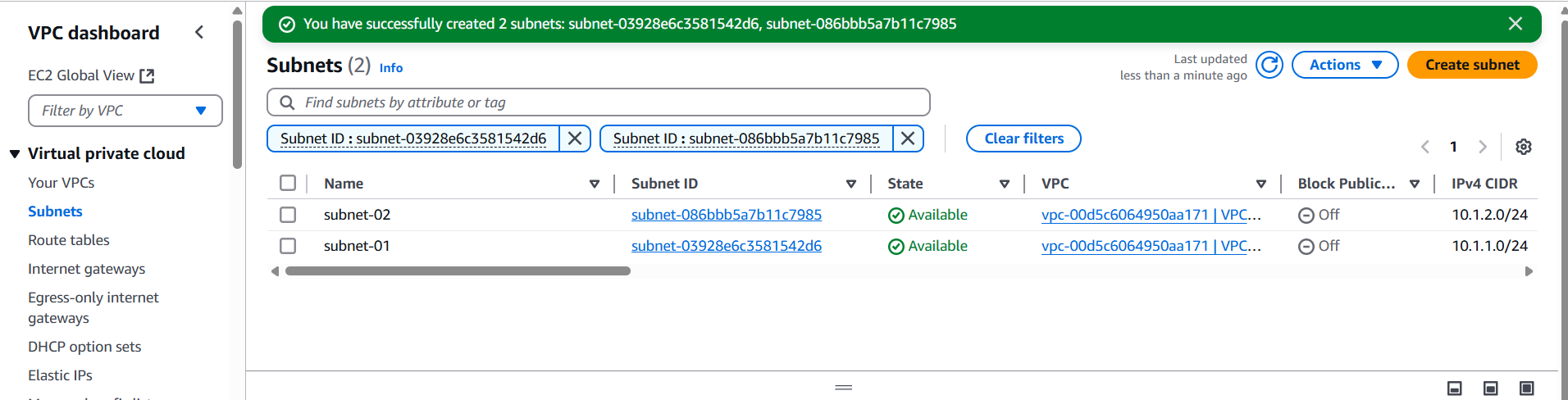


Fig: Two subnets (subnet-01 & subnet-02).

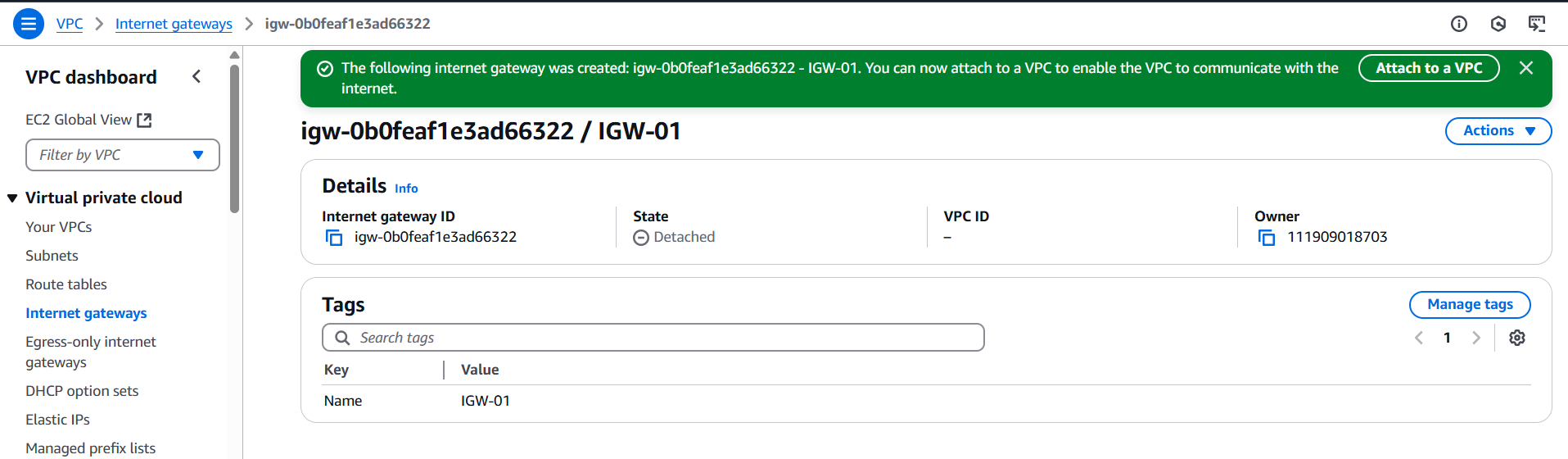
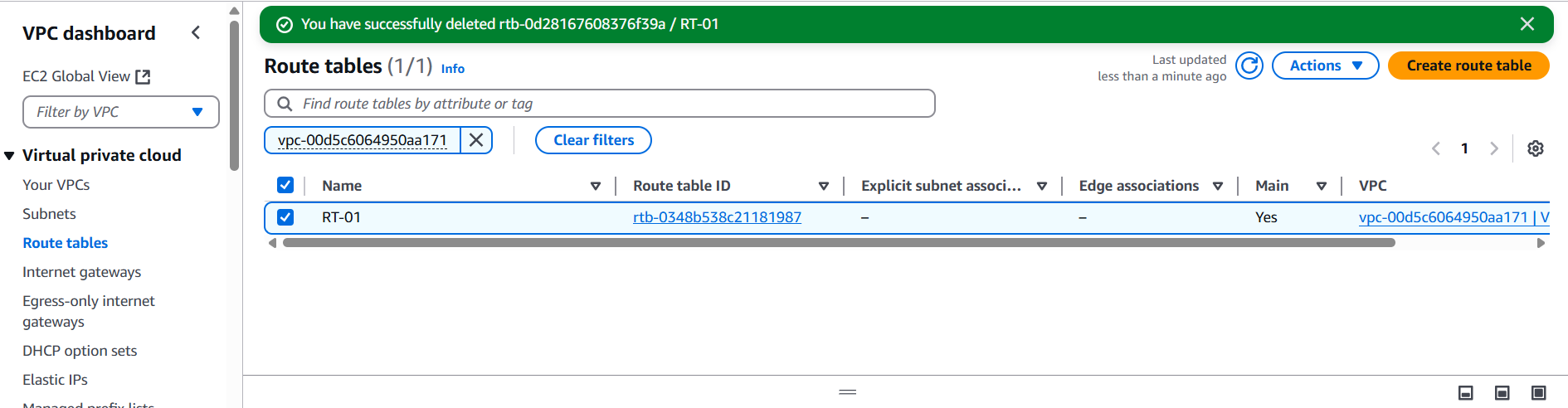


Fig: Internet gateway (IGW-01). Attach it to the VPC-01



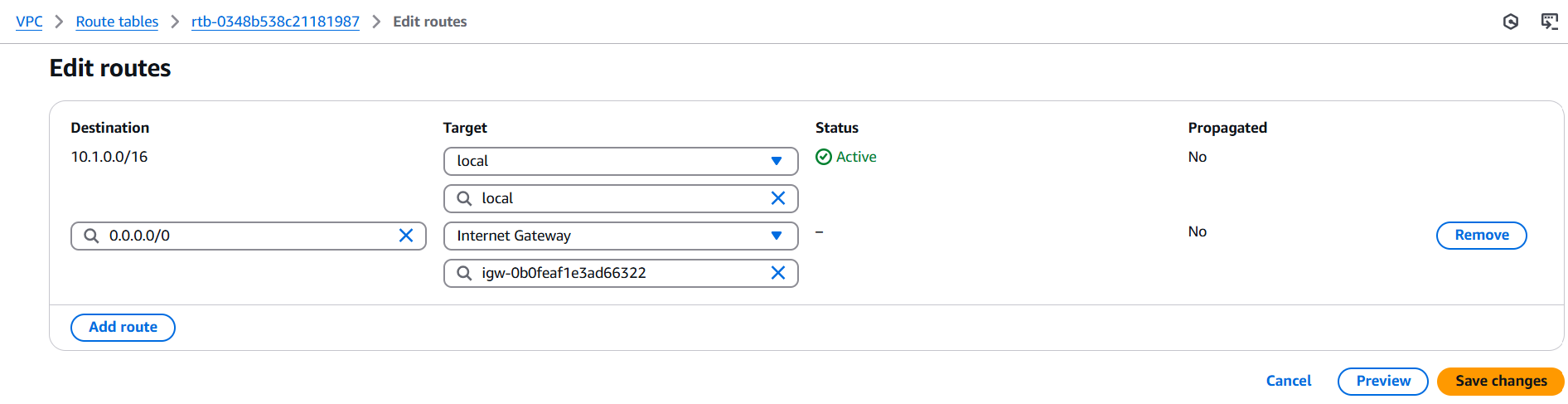


Fig: Configuring Route table.

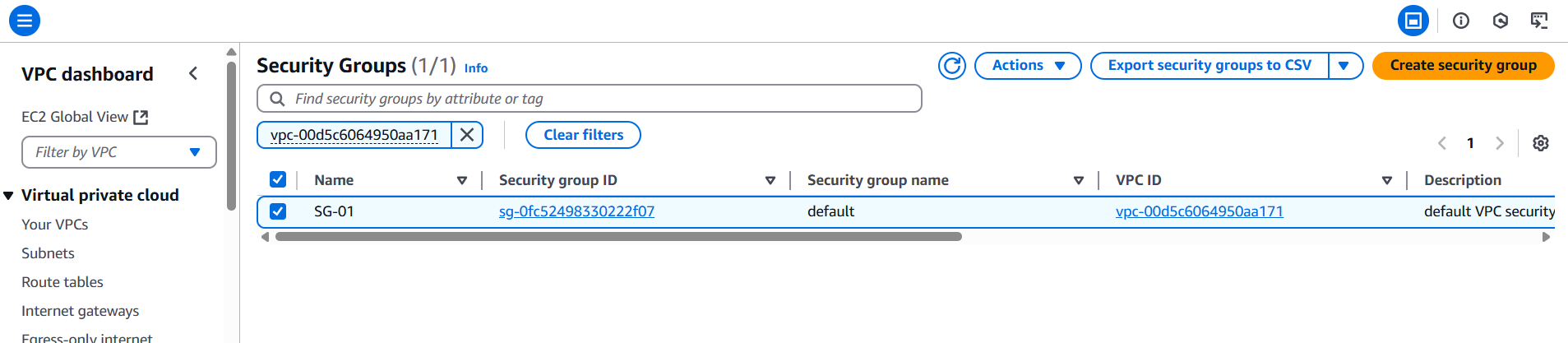


Fig: Security group (SG-01).

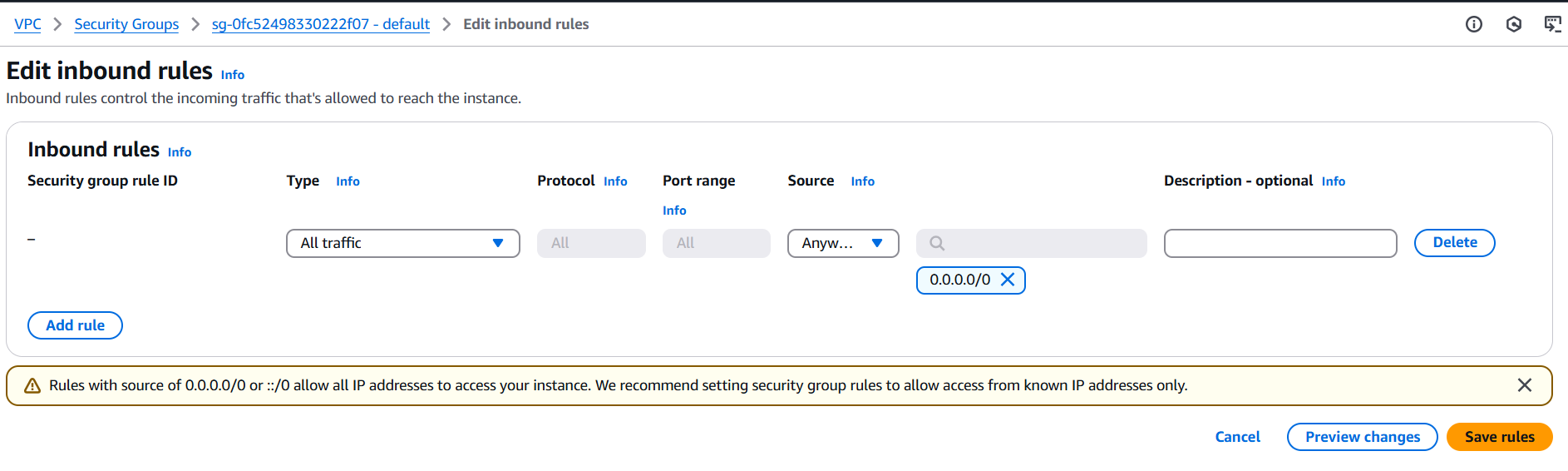


Fig: SG inbound rules.

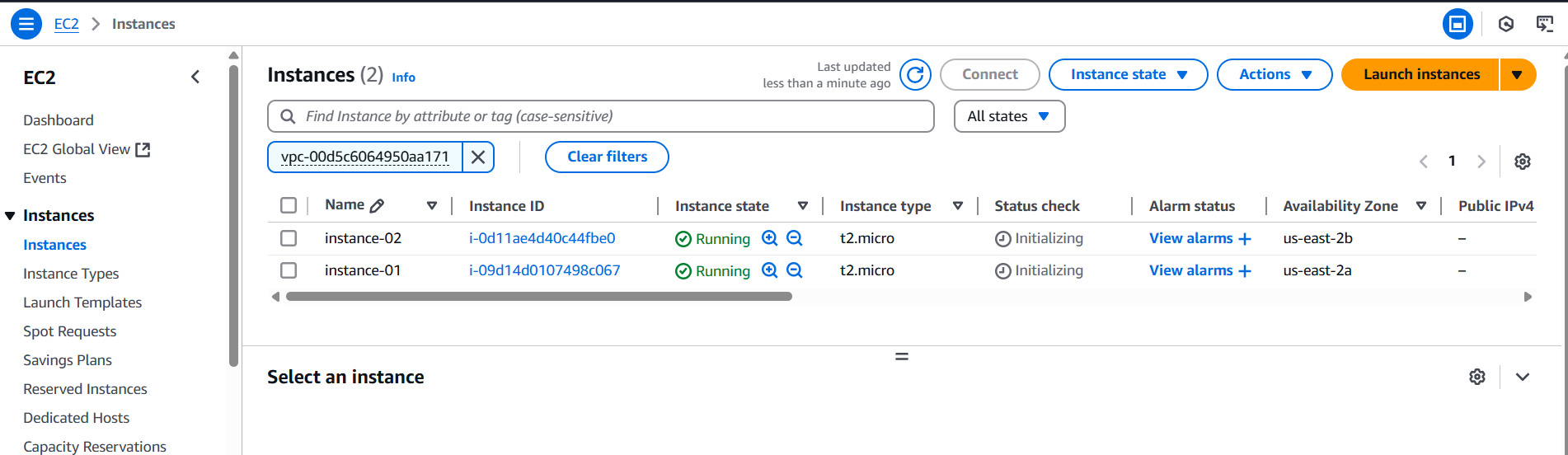


Fig: Two EC2 instances (Instance-01 & instance-02) are created successfully.

**Step2:** Now install the web server Nginx in both the instances (Instnace-01& Instances-02).

Installing of Nginx server:

1. sudo su
2. apt update
3. apt install nginx -y

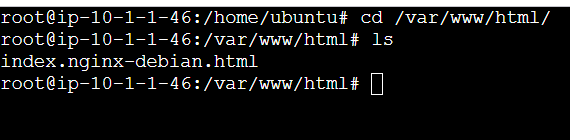
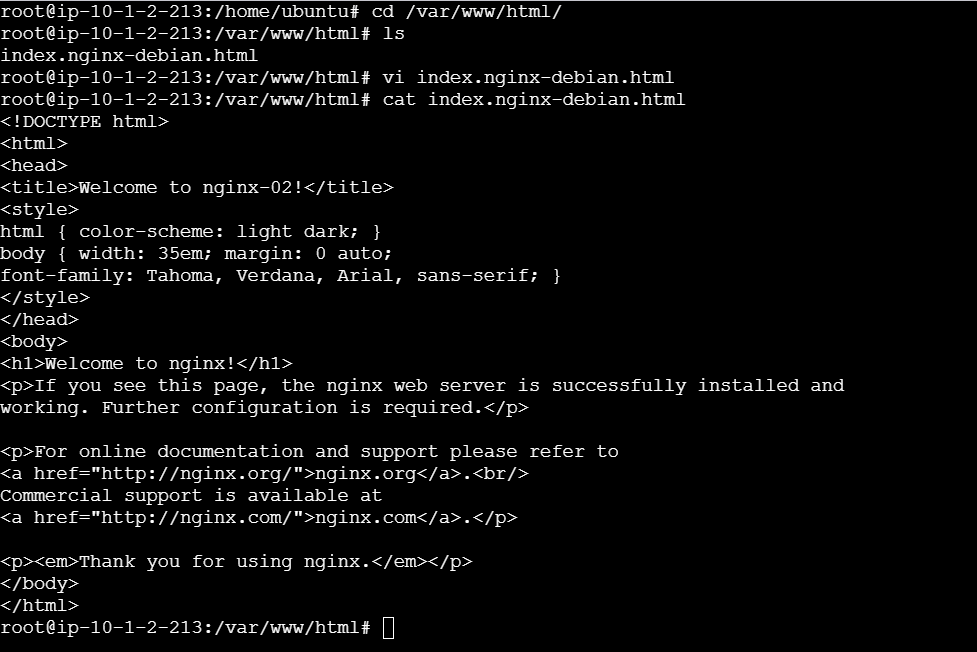


Fig: Nginx web server in instance-01.



**Fig:** Nginx web server in instance-02

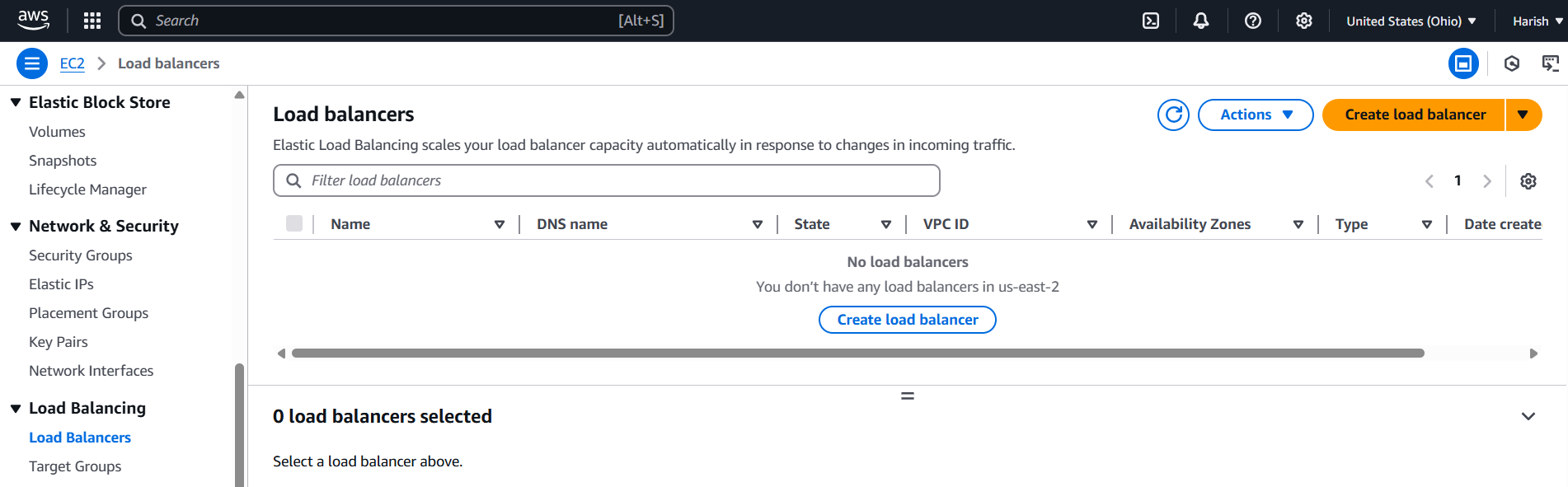
In instance-02 a small modification is done (Welcome to nginx-02) in its existing content using VI editor

**Command:** vi index.nginx-debian.html

Press “I”🡺To edit content in it

Press “Esc” and enter “:wq”🡺 It will save and exit from VI editor.

**Step3:** Create a Network Load Balancer (NLB-01).



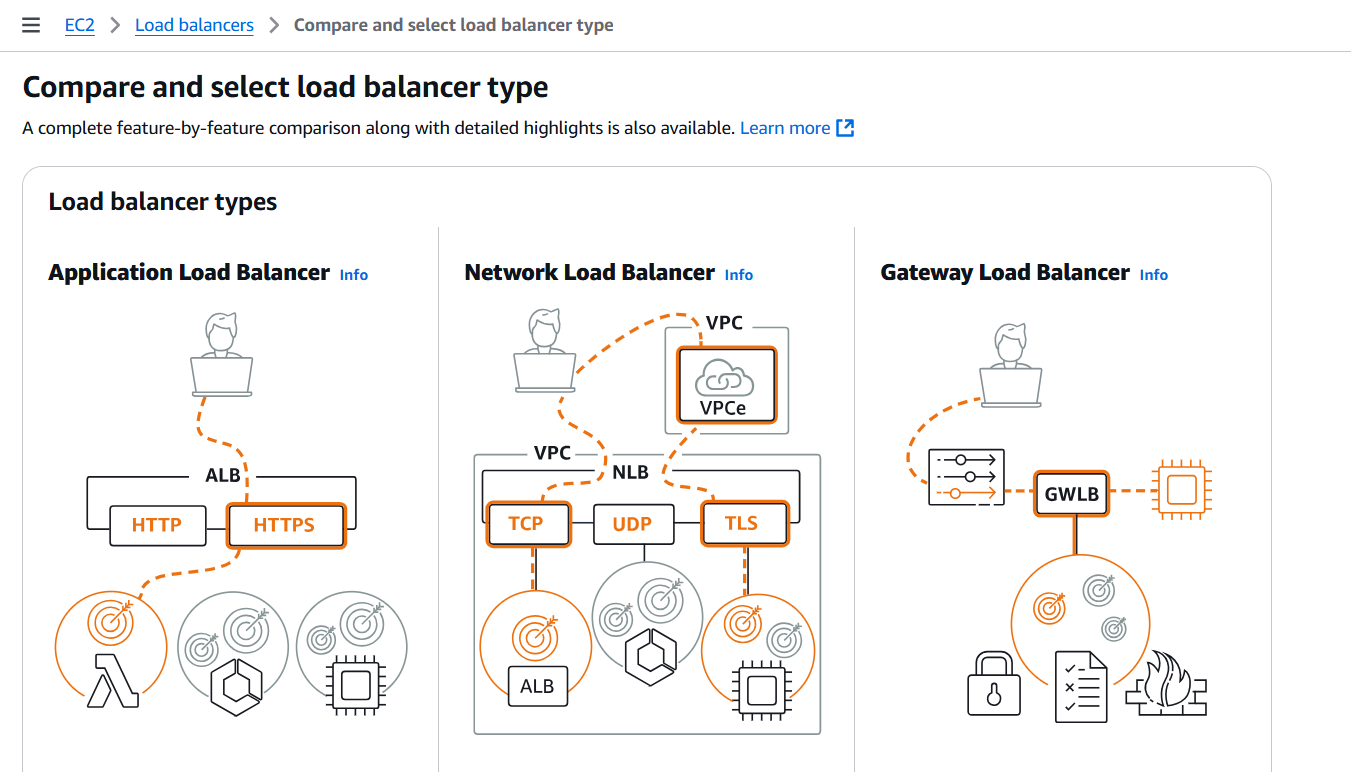


Fig: Network Load Balancer.

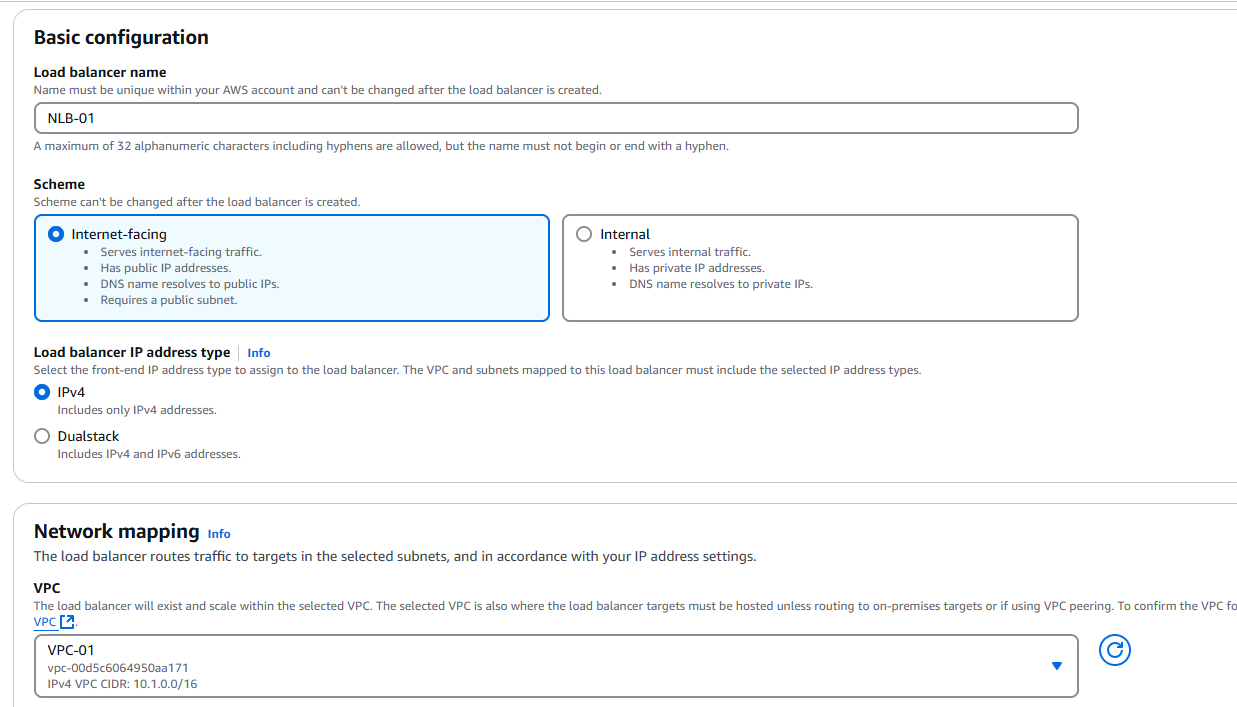


Fig: Configuring of Load balancer.

Internet facing 🡪 Public Network Load balancer.

Internal 🡪 Private Network Load balancer.

Now configure the target group (Backend pool) for the Network Load Balancer.

In target group we can choose/add instances, IP address, Lambda functions and Application Load Balancer.

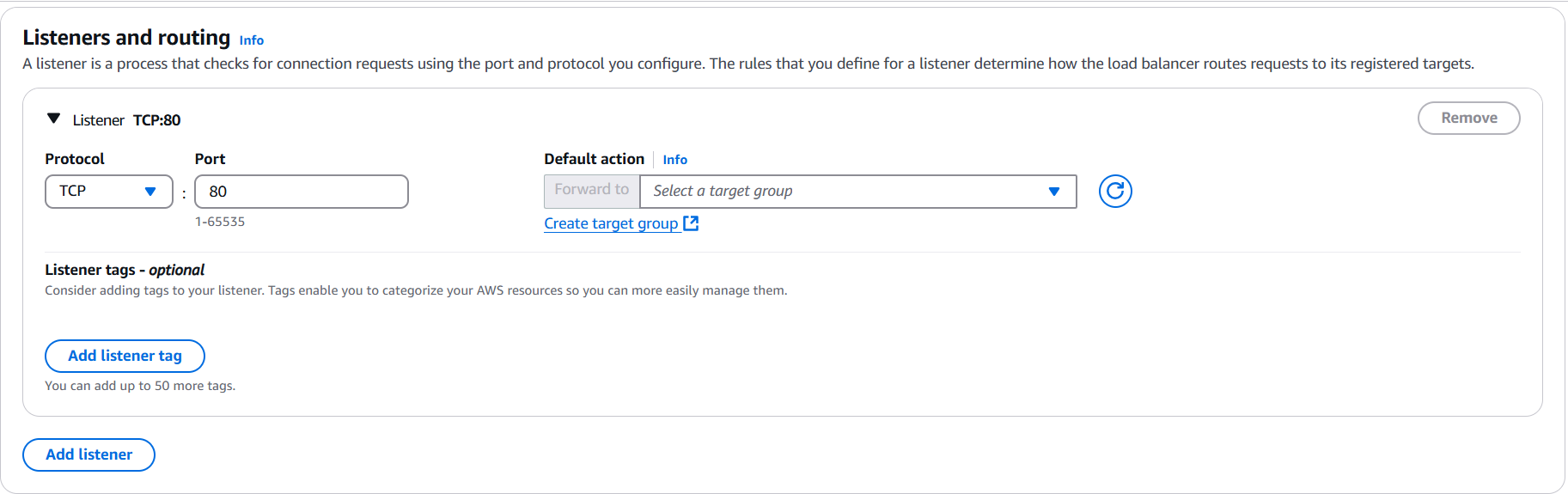


Fig: Target Group.

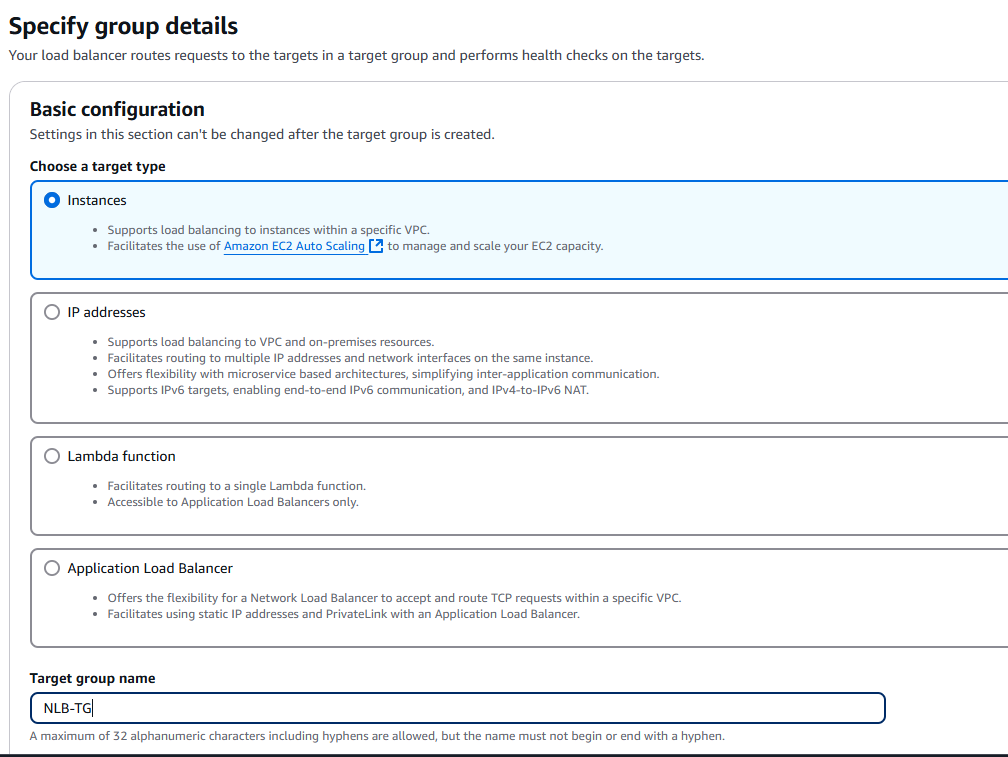


Fig: Configuring of target group.

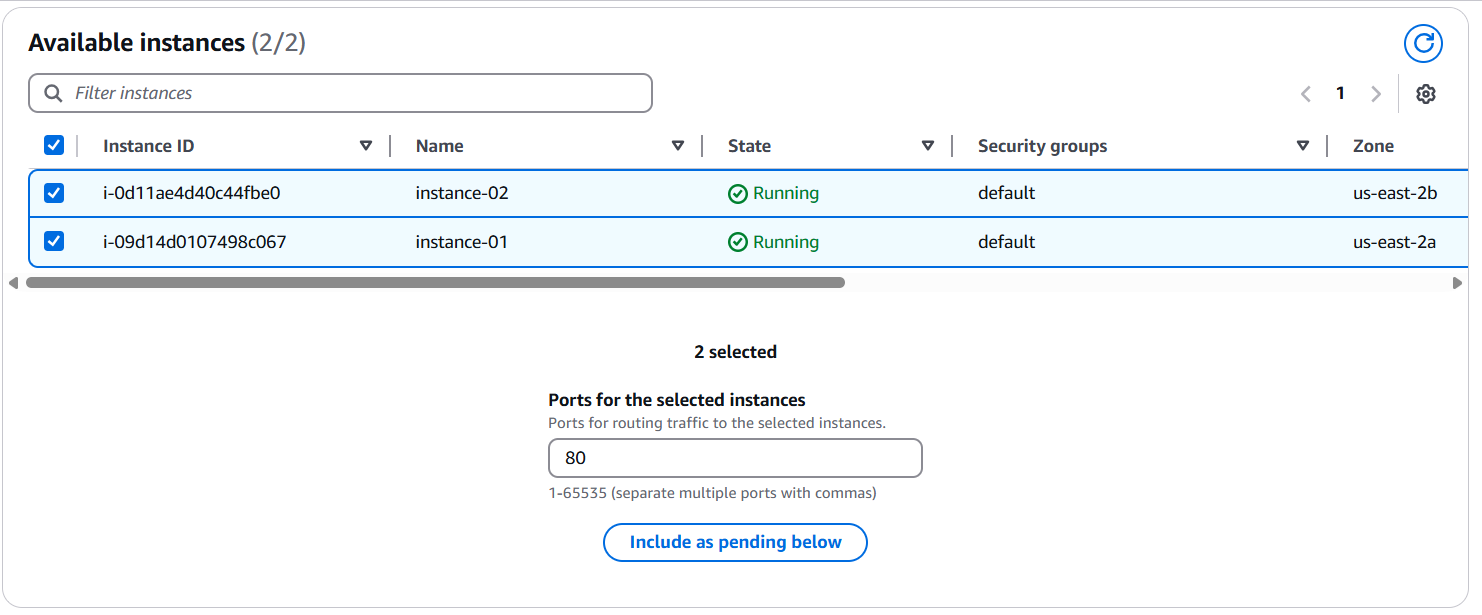


Fig: Adding of two instances (Instance-01 & Instance-02) to the target group (NLB-TG).

**Note:** Click “**Include as pending below**” option because the selected targets are listed before finalizing the target group.

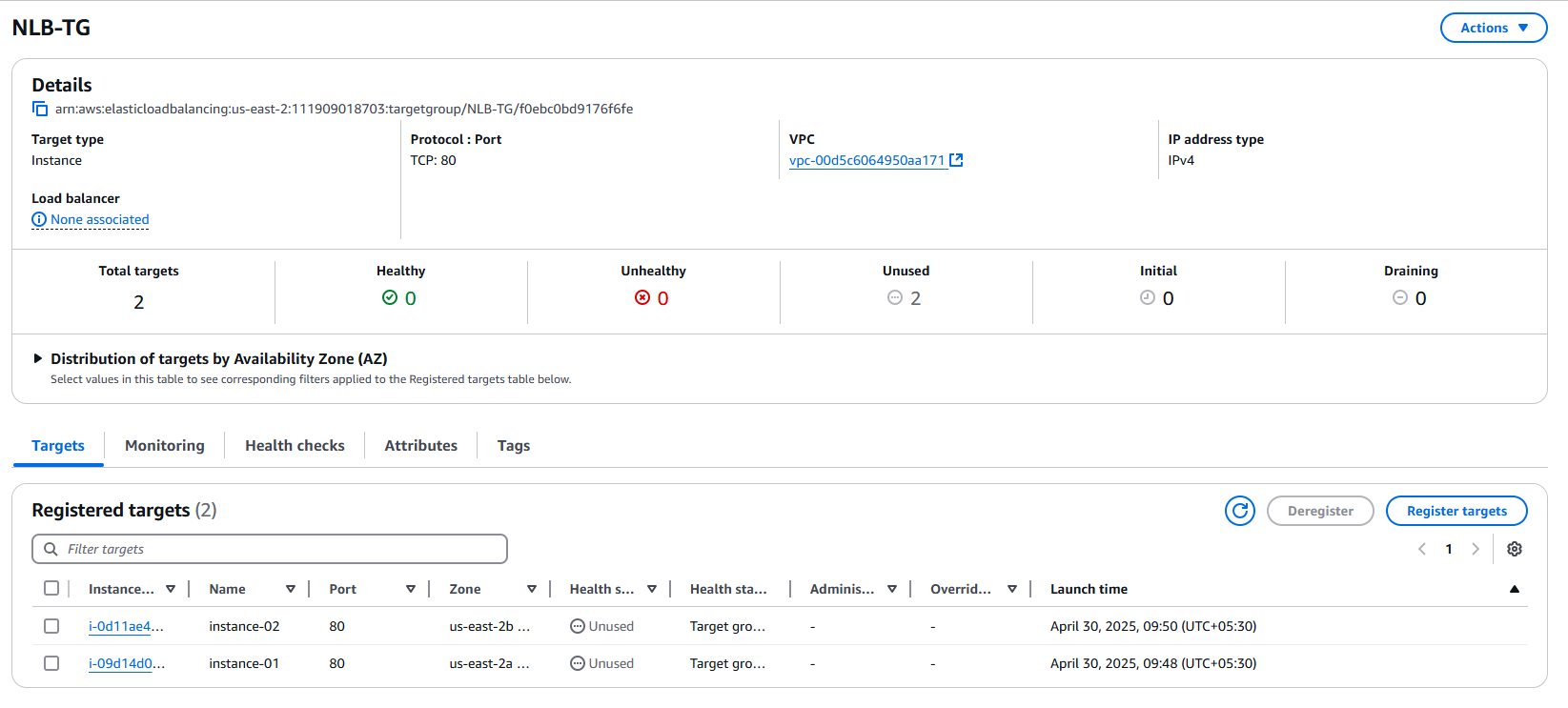


Fig: Target group (NLB-TG) is created successfully.

Now select the target group (NLB-TG) in the configuration of Network load balancer.

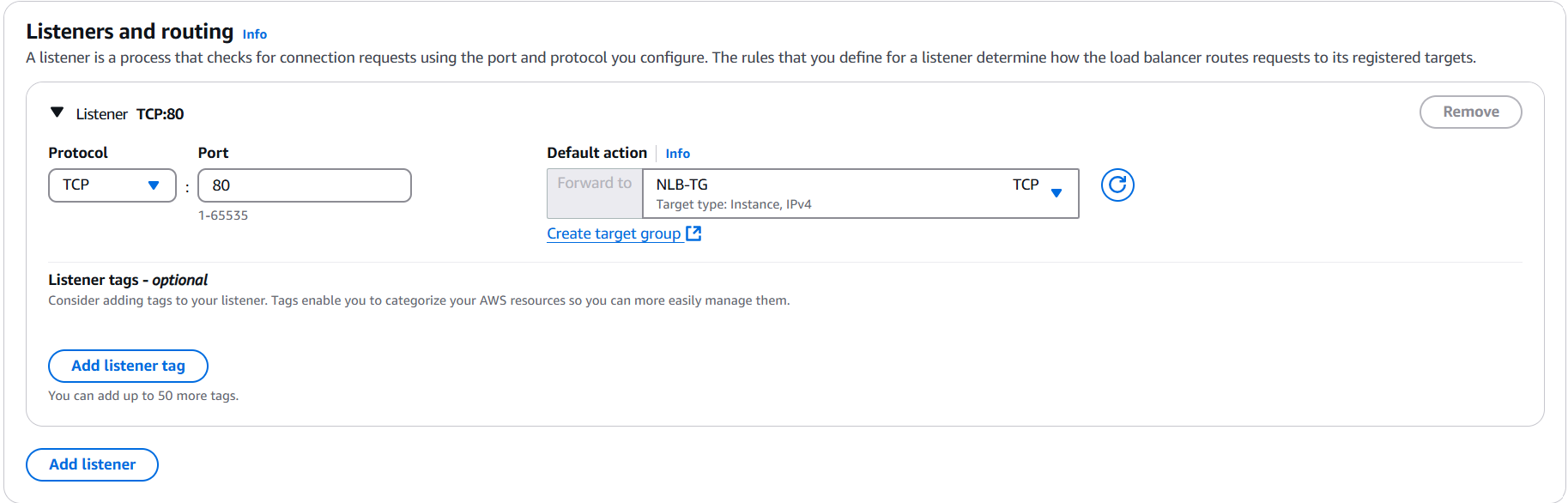


Fig: Target Group (NLB-TG) is selected.

Now create the Network load balancer.

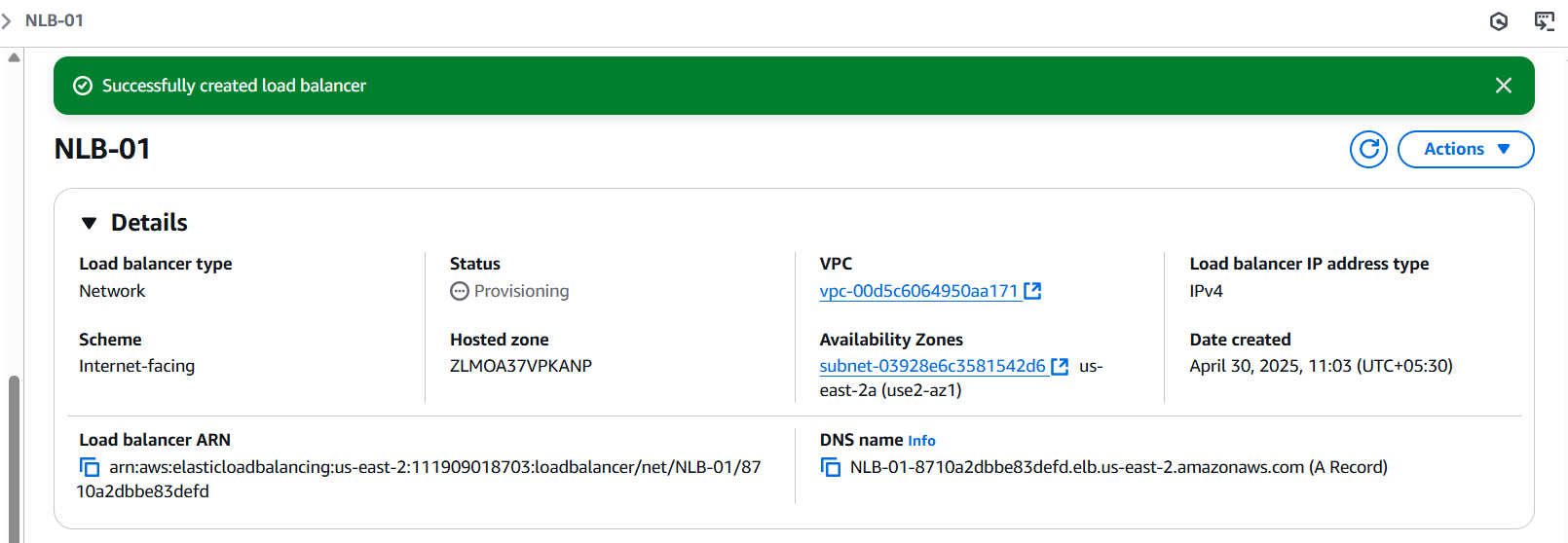


Fig: Network load balancer is created successfully.

By using the DNS name provided above, we can browse our application directly.

To change/configure the Domain name (Host A record/ C name record) AWS provide a service called “Rout 53” which is same as “Azure DNS”, by using Route 53 service we can perform ‘**Host A record**” and “**C name record”.**

**Step4:** Now copy the (Public IP) DNS name of Network load balancer and brows it.

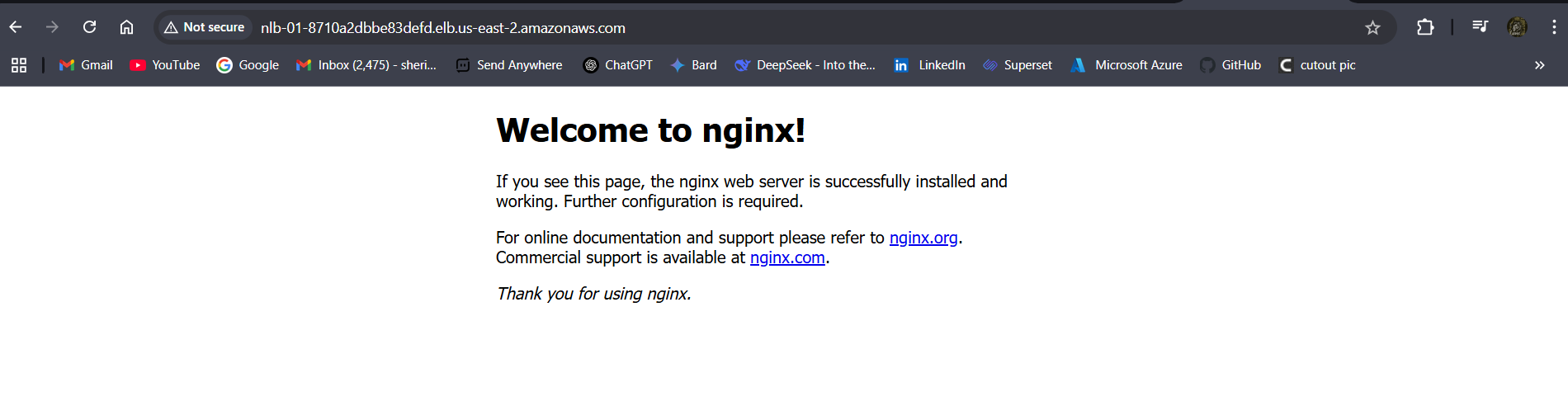


Fig: Nginx Web page.

By using the DNS name of Network Load balancer we browsed the Nginx web page successfully.

**Note:** In AWS by Using DNS name of Load balancer only, we can browse. Not public IP.

**Note2:** We can decode the DNS name to get Public IP address of Network load balancer, the command is

**Command:** nslookup <DNS name of Load Balancer> (perform it in Linux instances or bash or cmd or power shell )

**EX:** nslookup NLB-01-8710a2dbbe83defd.elb.us-east-2.amazonaws.com

**Block Diagram:**

Go daddy

Rout 53

Host A record

Name server-1

Name server-2

Name server-3

Name server-4

Name server-1

Name server-2

Name server-3

Name server-4

NLB

Subnet-02

Subnet-01

VPC-01

Instance-02

Client/user

Machine

Updating of name servers in GoDaddy

Instance-01