A SUMMER INTERSHIP REPORT ON

“DEMONSTRATE LOAD BALANCER BALANCING LOAD IN DIFFERENT REGIONS”



DEPARTMENT OF COMPUTER ENGINEERING & APPLICATIONS

INSTITUTE OF ENGINEERING & TECHNOLOGY

GLA UNIVERSITY, MATHURA

**BATCH 2022-2026**

**SUBMITTED BY: SUPERVISED BY:**

GAGAN SINGH (2215000657) RAUSHAN

DEEPAK GAUTAM (2215000543) KUMAR SINGH

ROBIN CHHONKAR (2215001483)

TUSHAR SHRIVASTAVA (2215001869)

ARMAN AHMED (2215000341)

**Declaration**

We are the student of B.tech (V Semester) Session 2024-2025, Batch 2022-2026 hereby declare that my work entitled “Load Balancer ”, is the outcome of genuine efforts done by me under the able guidance of Raushan Kumar Singh and being submitted to “Institute of Engineering & Technology”, GLA University, Mathura as summer training project report in partial fulfilment for the award of the degree of Bachelor of Technology (B.tech).

Place : Mathura

Date : 14-08-2024

Course : B.tech (V Semester)

Name :

GAGAN SINGH (2215000657)

DEEPAK GAUTAM (2215000543)

ROBIN CHHONKAR (2215001483)

TUSHAR SHRIVASTAVA (2215001869)

ARMAN AHMED (2215000341)

CONTENT

1. Introduction……………………………………………4
2. How does load balancing work……………………5-7

Types of load balancing and its technology…….8-10

AWS help in load balancing…………………………11

Create application load balancer on AWS……….12-19

Conclusion………………………………………………21-22

INTRODUCTION

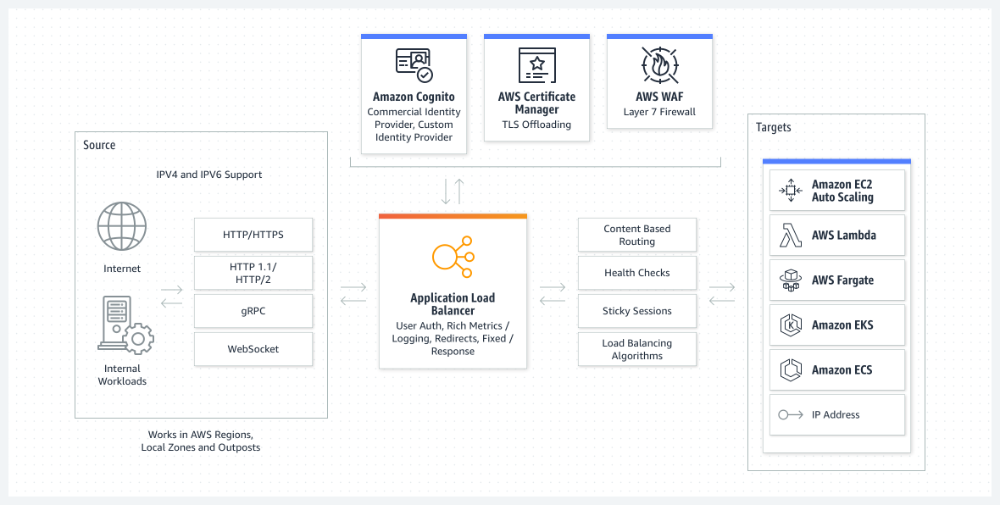
Load balancing is the method of distributing network traffic equally across a pool of resources that support an application. Modern applications must process millions of users simultaneously and return the correct text, videos, images, and other data to each user in a fast and reliable manner. To handle such high volumes of traffic, most applications have many resource servers with duplicate data between them. A load balancer is a device that sits between the user and the server group and acts as an invisible facilitator, ensuring that all resource servers are used equally. Load balancing is the practice of distributing computational workloads between two or more computers. On the Internet, load balancing is often employed to divide network traffic among several servers. This reduces the strain on each server and makes the servers more efficient, speeding up performance and reducing latency. Load balancing is essential for most Internet applications to function properly. Imagine a checkout line at a grocery store with 8 checkout lines, only one of which is open. All customers must get into the same line, and therefore it takes a long time for a customer to finish paying for their groceries. Now imagine that the store instead opens all 8 checkout lines. In this case, the wait time for customers is about 8 times shorter (depending on factors like how much food each customer is buying). Load balancing essentially accomplishes the same thing. This results in a better user experience — the grocery store customers in the example above would probably look for a more efficient grocery store if they always experienced long wait times.

LOAD BALANCING WORK AND ITS ALGORITHM

**Work of load balancing:-**

Companies usually have their application running on multiple servers. Such a server arrangement is called a server farm. User requests to the application first go to the load balancer. The load balancer then routes each request to a single server in the server farm best suited to handle the request.

Load balancing is like the work done by a manager in a restaurant. Consider a restaurant with five waiters. If customers were allowed to choose their waiters, one or two waiters could be overloaded with work while the others are idle. To avoid this scenario, the restaurant manager assigns customers to the specific waiters who are best suited to serve them.



**Algorithms:-**

A load balancing algorithm is the set of rules that a load balancer follows to determine the best server for each of the different client requests. Load balancing algorithms fall into two main categories.

**Static load balancing**

Static load balancing algorithms follow fixed rules and are independent of the current server state. The following are examples of static load balancing.

***Round-robin method***

Servers have IP addresses that tell the client where to send requests. The IP address is a long number that is difficult to remember. To make it easy, a Domain Name System maps website names to servers. When you enter [aws.amazon.com](https://aws.amazon.com/) into your browser, the request first goes to our name server, which returns our IP address to your browser.

In the round-robin method, an authoritative name server does the load balancing instead of specialized hardware or software. The name server returns the IP addresses of different servers in the server farm turn by turn or in a round-robin fashion.

***Weighted round-robin method***

In weighted round-robin load balancing, you can assign different weights to each server based on their priority or capacity. Servers with higher weights will receive more incoming application traffic from the name server.

***IP hash method***

In the IP hash method, the load balancer performs a mathematical computation, called hashing, on the client IP address. It converts the client IP address to a number, which is then mapped to individual servers.

**Dynamic load balancing**

Dynamic load balancing algorithms examine the current state of the servers before distributing traffic. The following are some examples of dynamic load balancing algorithms.

***Least connection method***

A connection is an open communication channel between a client and a server. When the client sends the first request to the server, they authenticate and establish an active connection between each other. In the least connection method, the load balancer checks which servers have the fewest active connections and sends traffic to those servers.

***Weighted least connection method***

Weighted least connection algorithms assume that some servers can handle more active connections than others. Therefore, you can assign different weights or capacities to each server, and the load balancer sends the new client requests to the server with the least connections by capacity.

***Least response time method***

The response time is the total time that the server takes to process the incoming requests and send a response. The least response time method combines the server response time and the active connections to determine the best server. Load balancers use this algorithm to ensure faster service for all users.

***Resource-based method***

In the resource-based method, load balancers distribute traffic by analyzing the current server load. Specialized software called an agent runs on each server and calculates usage of server resources, such as its computing capacity and memory. Then, the load balancer checks the agent for sufficient free resources before distributing traffic to that server.

TYPES OF LOAD BALANCING AND ITS TECHNOLOGY

**Types of load balancing:-**

We can classify load balancing into three main categories depending on what the load balancer checks in the client request to redirect the traffic.

**Application load balancing**

Complex modern applications have several server farms with multiple servers dedicated to a single application function. Application load balancers look at the request content, such as HTTP headers or SSL session IDs, to redirect traffic.

For example, an ecommerce application has a product directory, shopping cart, and checkout functions. The application load balancer sends requests for browsing products to servers that contain images and videos but do not need to maintain open connections. By comparison, it sends shopping cart requests to servers that can maintain many client connections and save cart data for a long time.

**Network load balancing**

Network load balancers examine IP addresses and other network information to redirect traffic optimally. They track the source of the application traffic and can assign a static IP address to several servers. Network load balancers use the static and dynamic load balancing algorithms described earlier to balance server load.

**Global server load balancing**

Global server load balancing occurs across several geographically distributed servers. For example, companies can have servers in multiple data centres, in different countries, and in third-party cloud providers around the globe. In this case, local load balancers manage the application load within a region or zone. They attempt to redirect traffic to a server destination that is geographically closer to the client. They might redirect traffic to servers outside the client’s geographic zone only in case of server failure.

**DNS load balancing**

In DNS load balancing, you configure your domain to route network requests across a pool of resources on your domain. A domain can correspond to a website, a mail system, a print server, or another service that is made accessible through the internet. DNS load balancing is helpful for maintaining application availability and balancing network traffic across a globally distributed pool of resources.

**What are the types of load balancing technology?**

Load balancers are one of two types: hardware load balancer and software load balancer.

**Hardware load balancers**

A hardware-based load balancer is a hardware appliance that can securely process and redirect gigabytes of traffic to hundreds of different servers. You can store it in your data centers and use virtualization to create multiple digital or virtual load balancers that you can centrally manage.

**Software load balancers**

Software-based load balancers are applications that perform all load balancing functions. You can install them on any server or access them as a fully managed third-party service.

**Comparison of hardware balancers to software load balancers**

Hardware load balancers require an initial investment, configuration, and ongoing maintenance. You might also not use them to full capacity, especially if you purchase one only to handle peak-time traffic spikes. If traffic volume increases suddenly beyond its current capacity, this will affect users until you can purchase and set up another load balancer.

In contrast, software-based load balancers are much more flexible. They can scale up or down easily and are more compatible with modern cloud computing environments. They also cost less to set up, manage, and use over time.

AWS HELP IN LOAD BALANCING

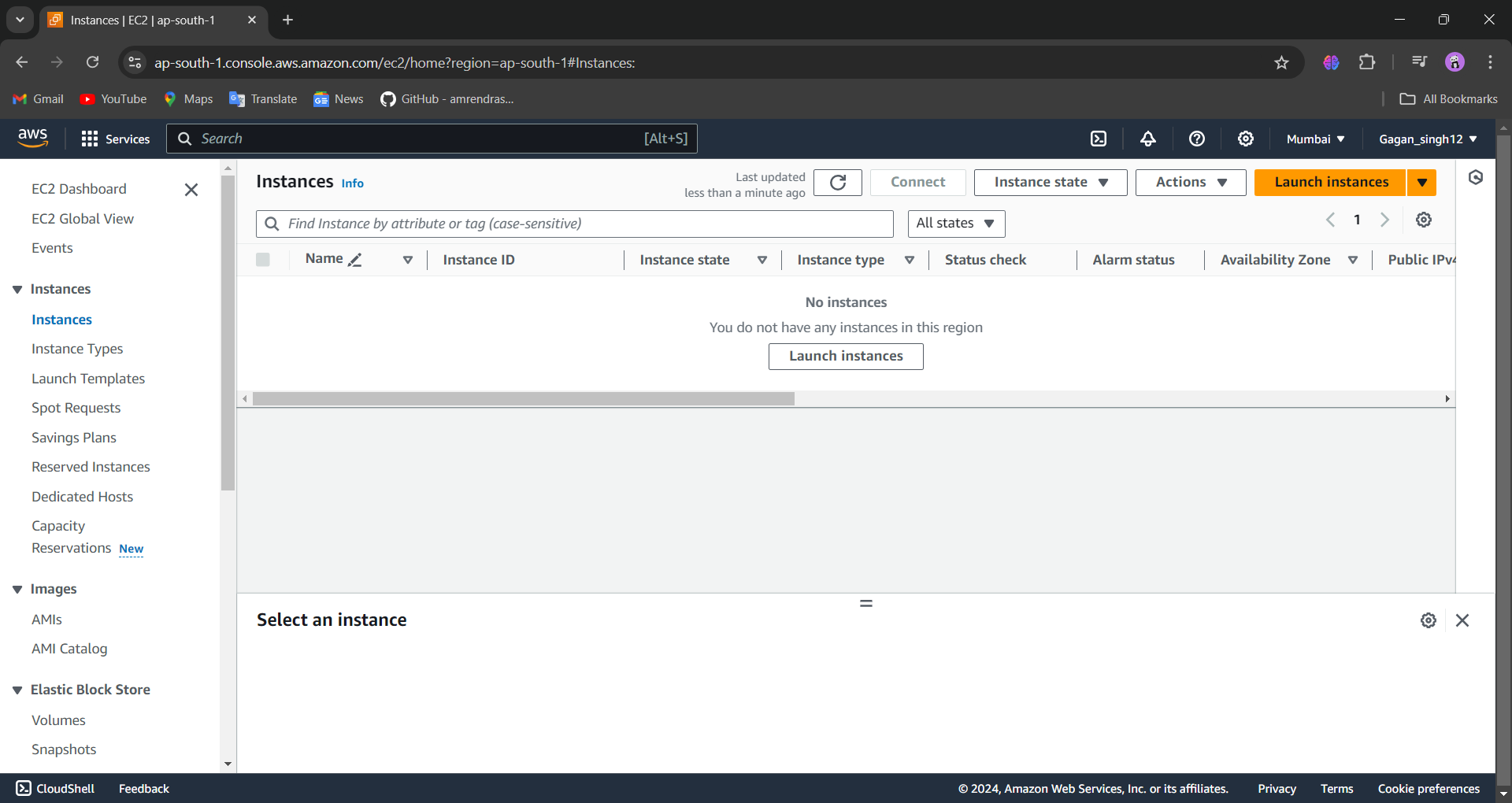
Elastic Load Balancing (ELB) is a fully managed load balancing service that automatically distributes incoming application traffic to multiple targets and virtual appliances across AWS and on-premises resources. You can use it to scale modern applications without complex configurations or API gateways. You can use ELB to set up four different types of software load balancers.

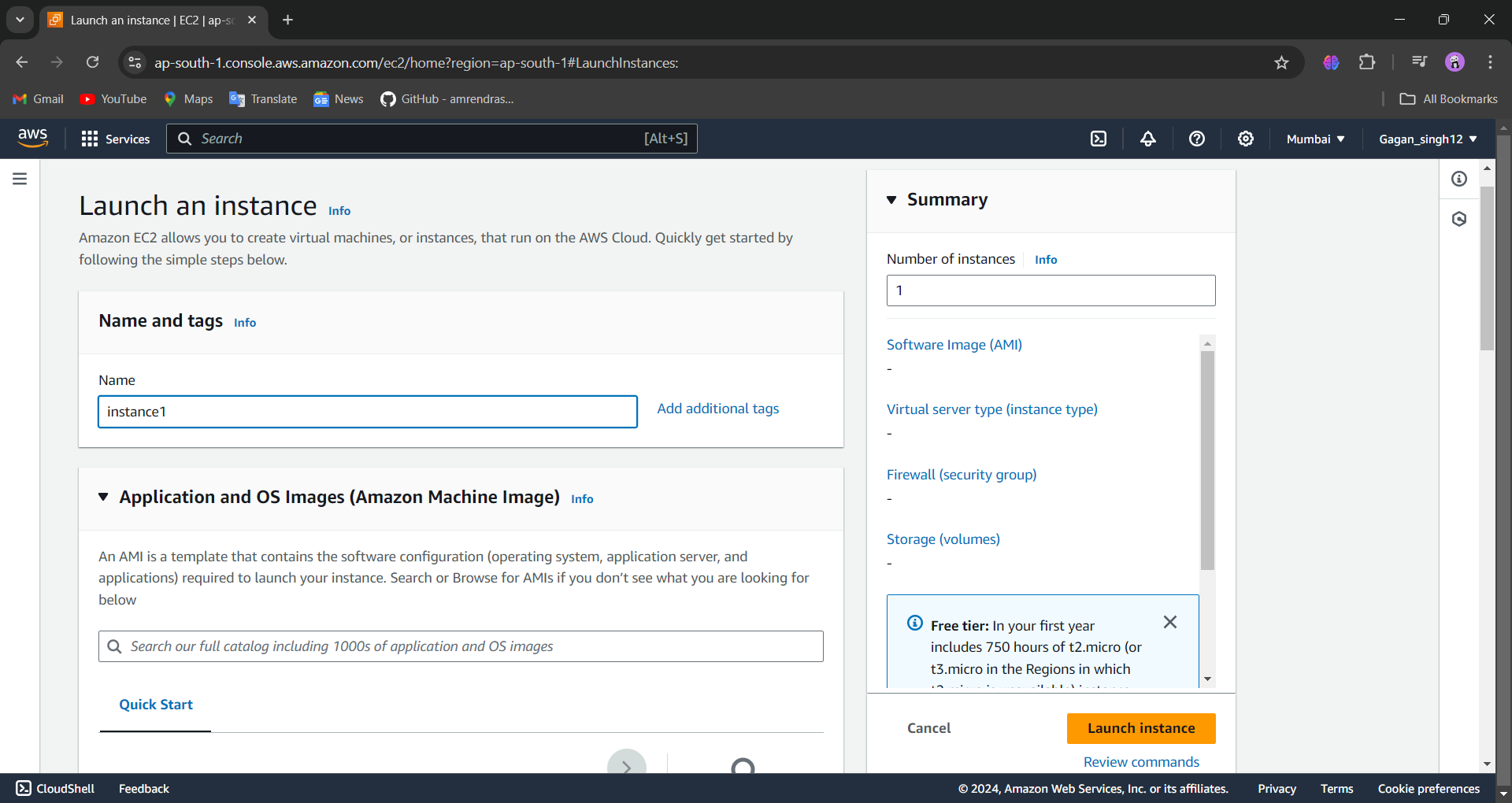
* An Application Load Balancer routes traffic for HTTP-based requests.
* A Network Load Balancer routes traffic based on IP addresses. It is ideal for balancing TCP and User Datagram Protocol (UDP)-based requests.
* A Gateway Load Balancer routes traffic to third-party virtual appliances. It is ideal for incorporating a third-party appliance, such as a network firewall, into your network traffic in a scalable and easy-to-manage way.
* A Classic Load Balancer routes traffic to applications in the Amazon EC2 -Classic network—a single, flat network that you share with other customers.

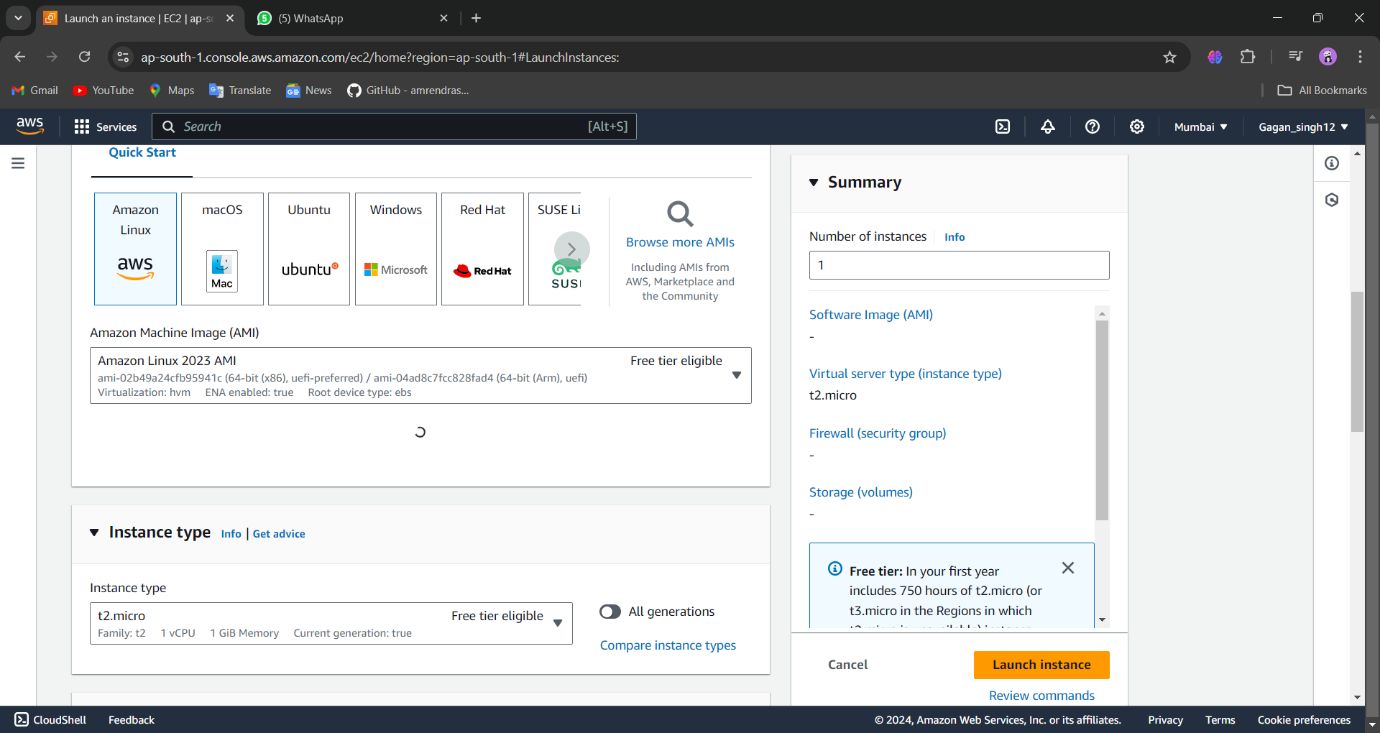
You can select the load balancer based on your requirements. For example, Terminix, a global pest control brand, uses Gateway Load Balancer to handle 300% more throughput. Second Spectrum, a company that provides artificial intelligence-driven tracking technology for sports broadcasts, uses AWS Load Balancer Controller to reduce hosting costs by 90%. Code.org, a nonprofit dedicated to expanding access to computer science in schools, uses Application Load Balancer to handle a 400% spike in traffic efficiently during online coding events.

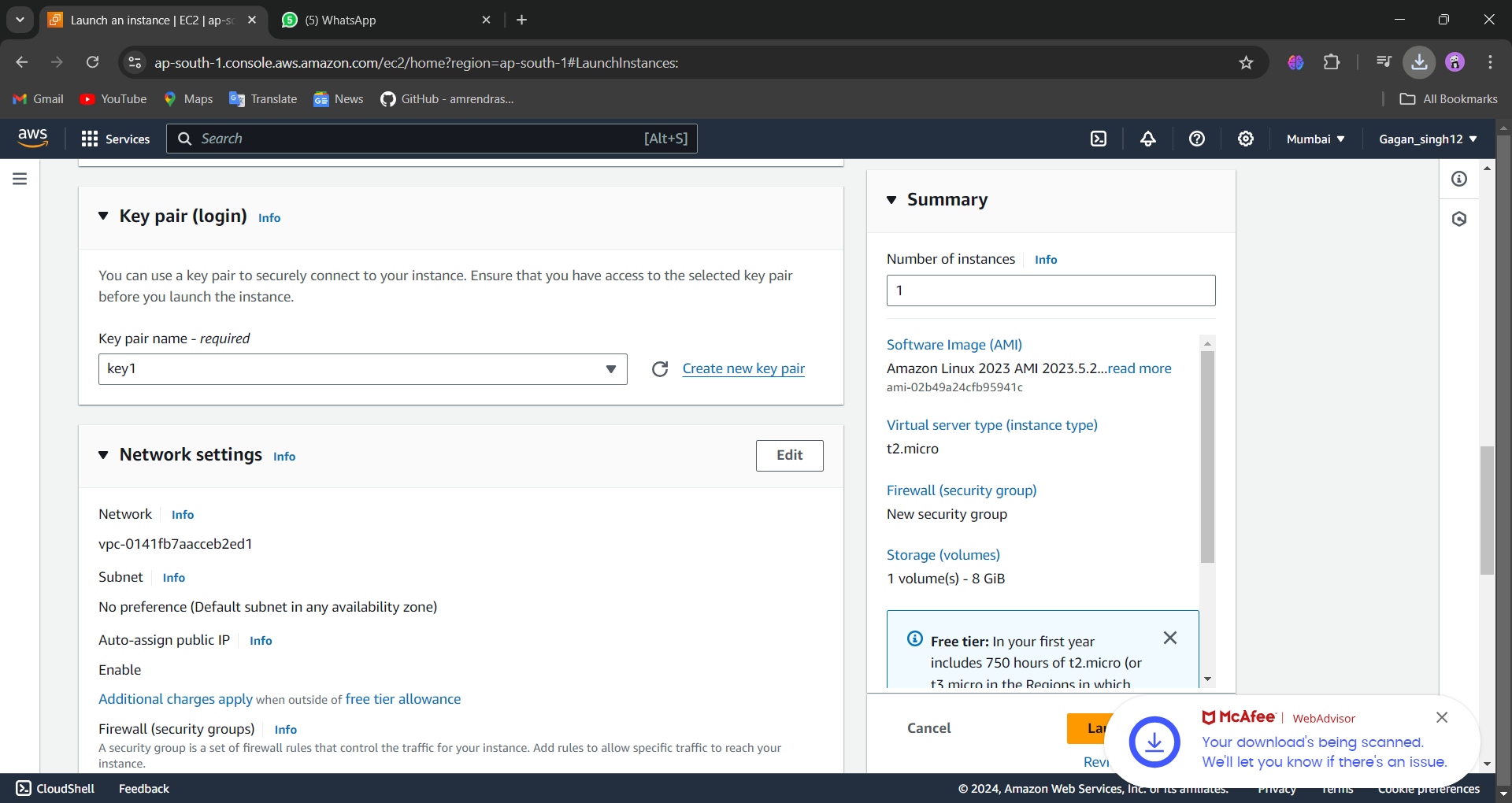
CREATE APPLICATION LOAD BALANCER ON AWS

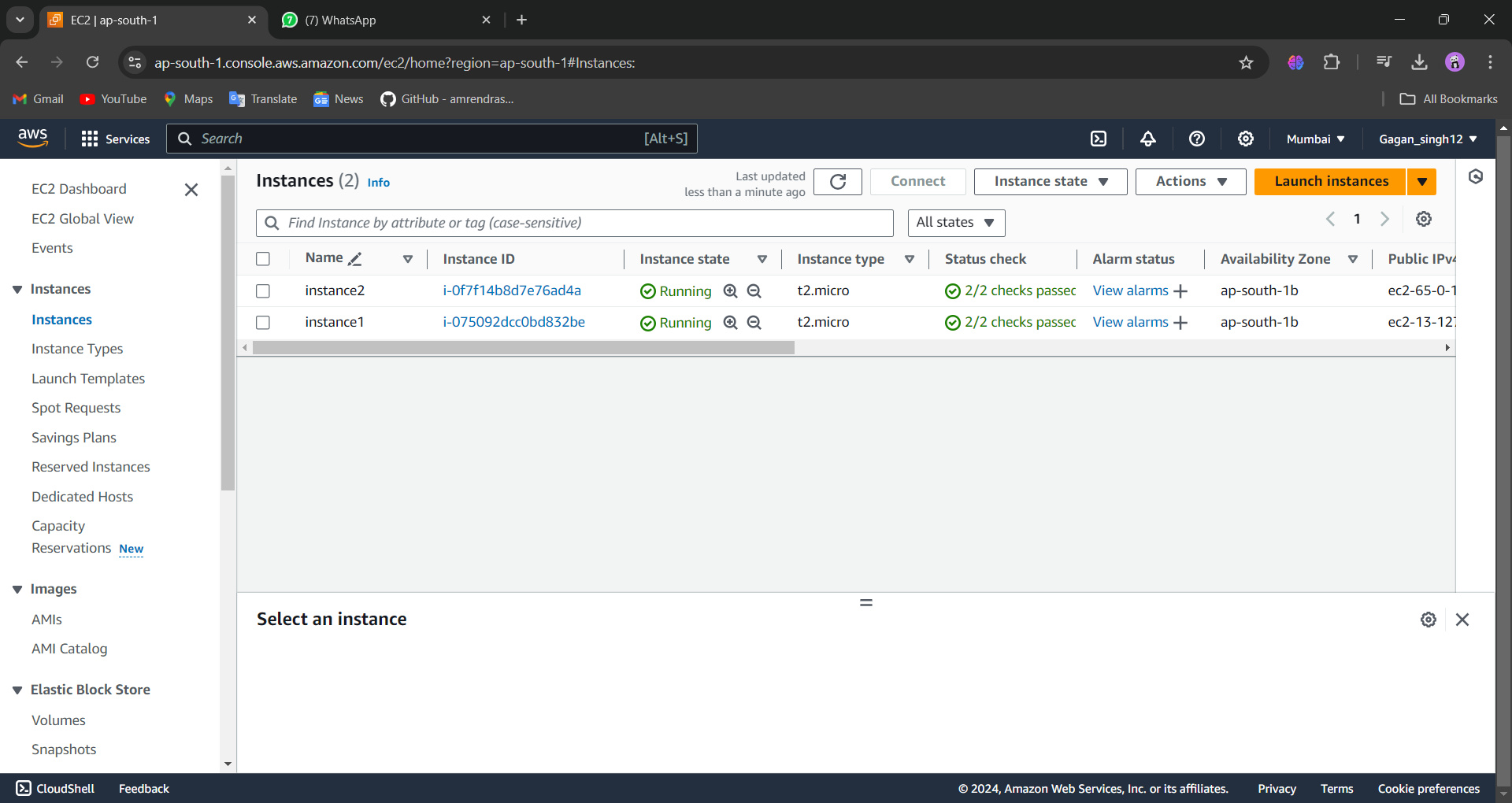
**First we’ll create EC2 instance:-**

****

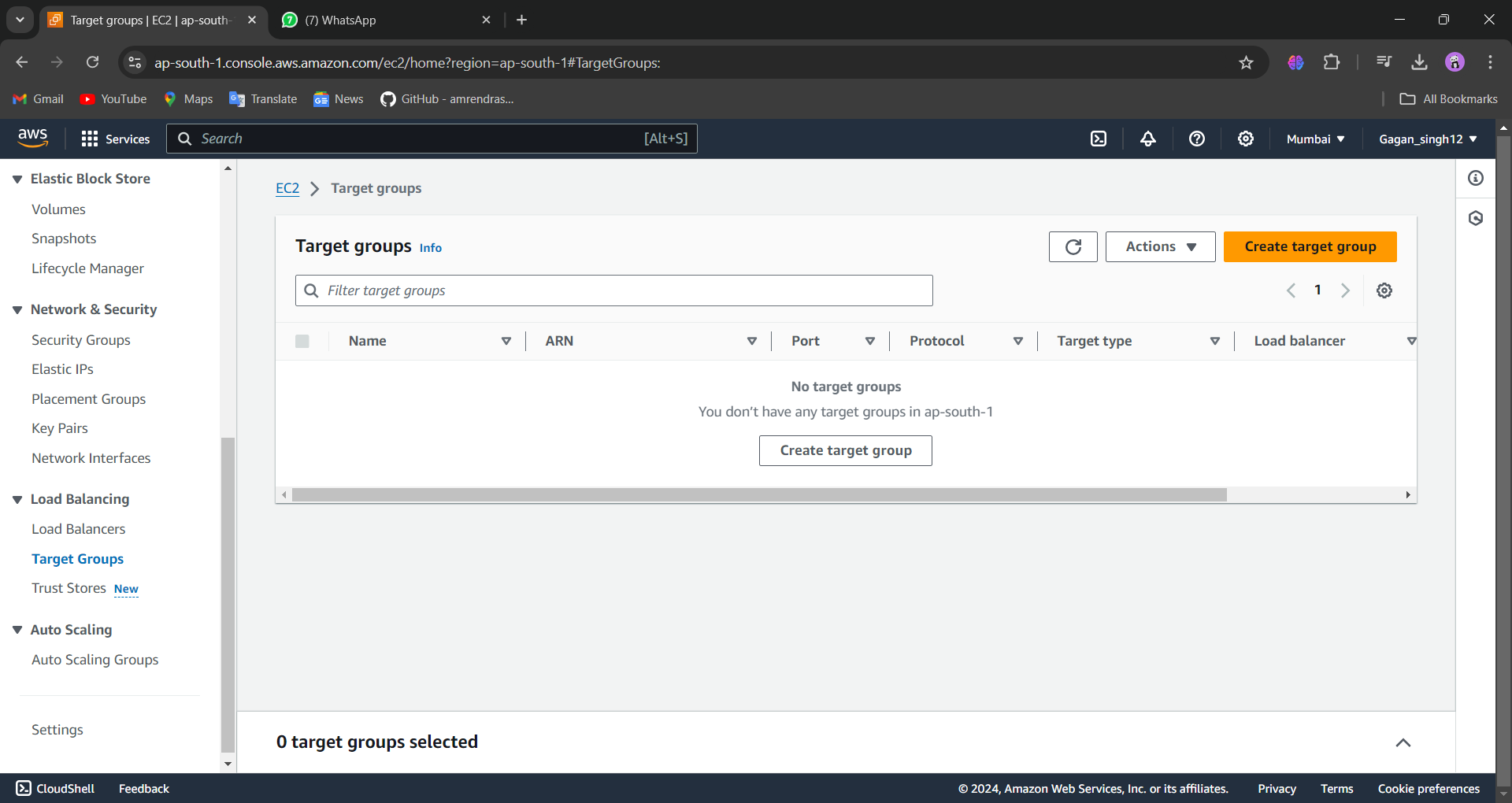
****

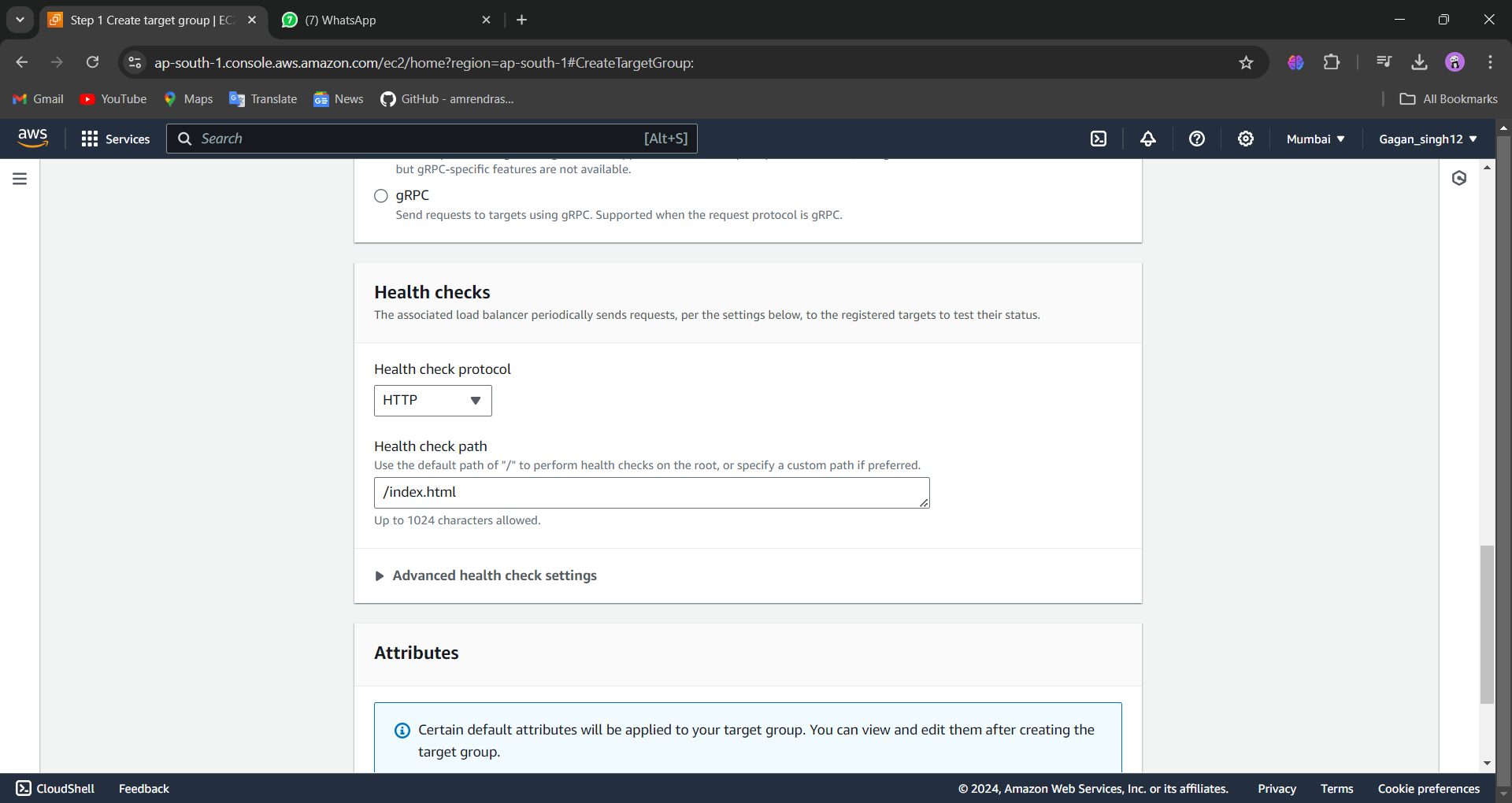
****

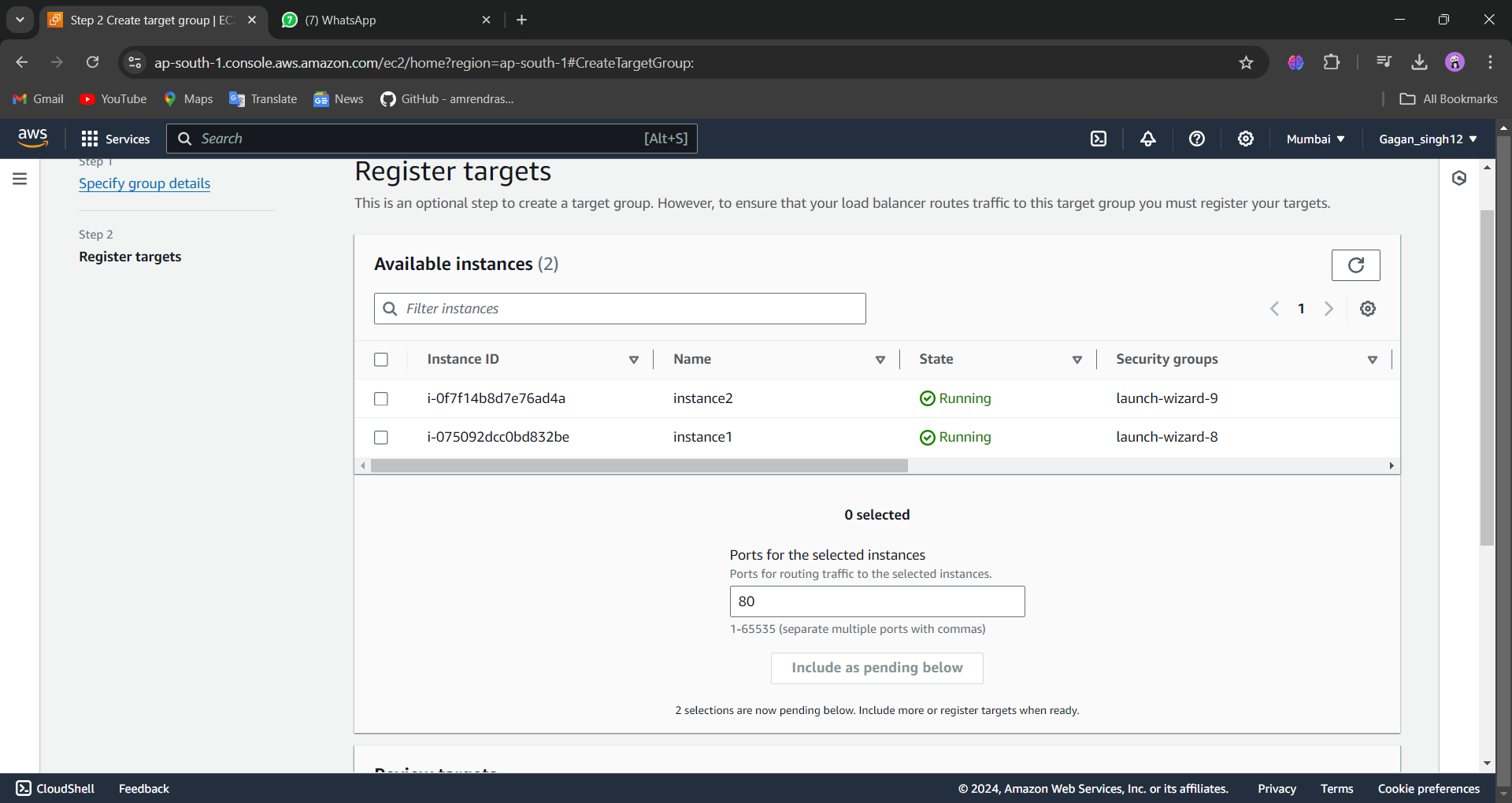
****

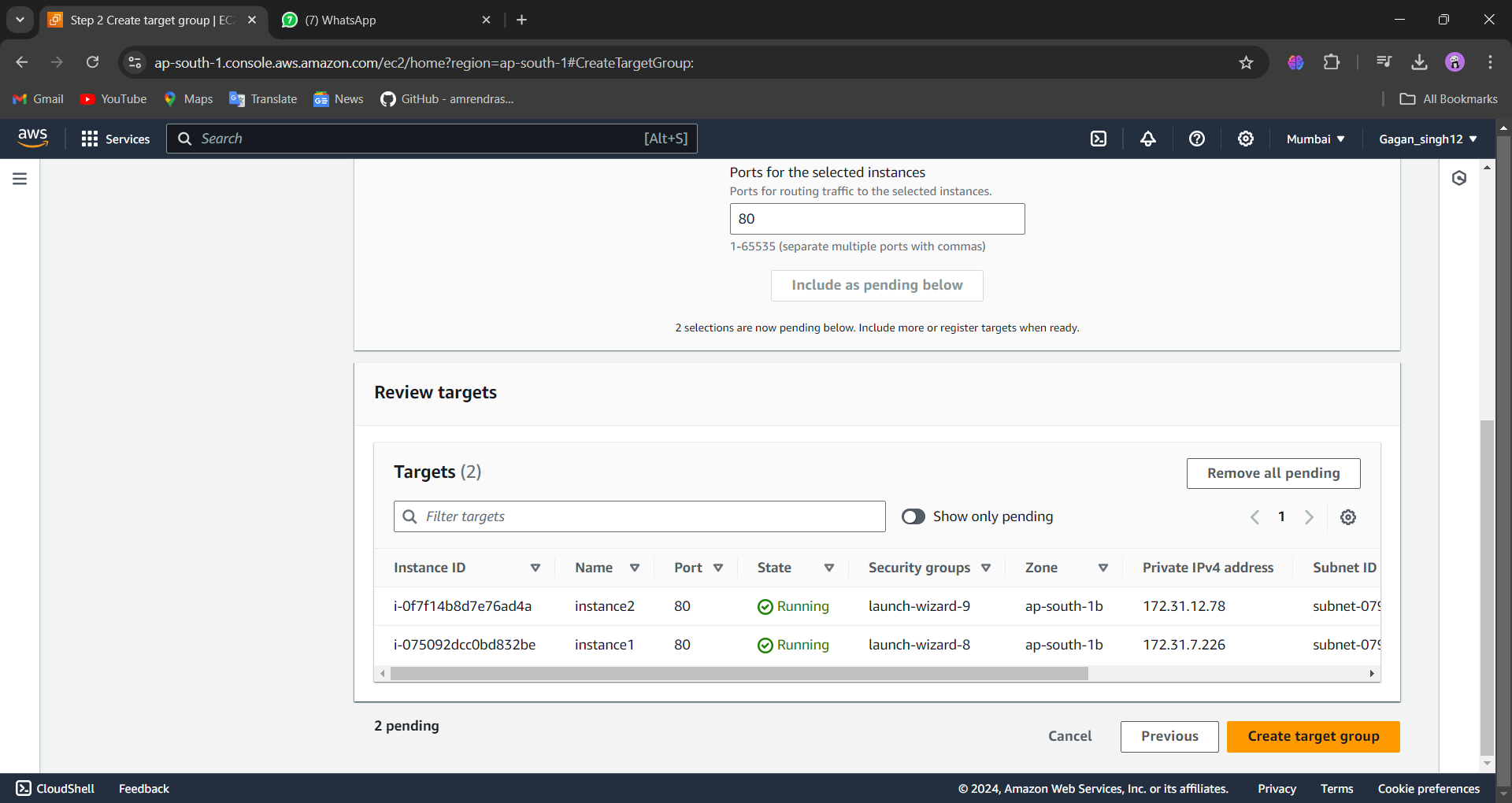
****

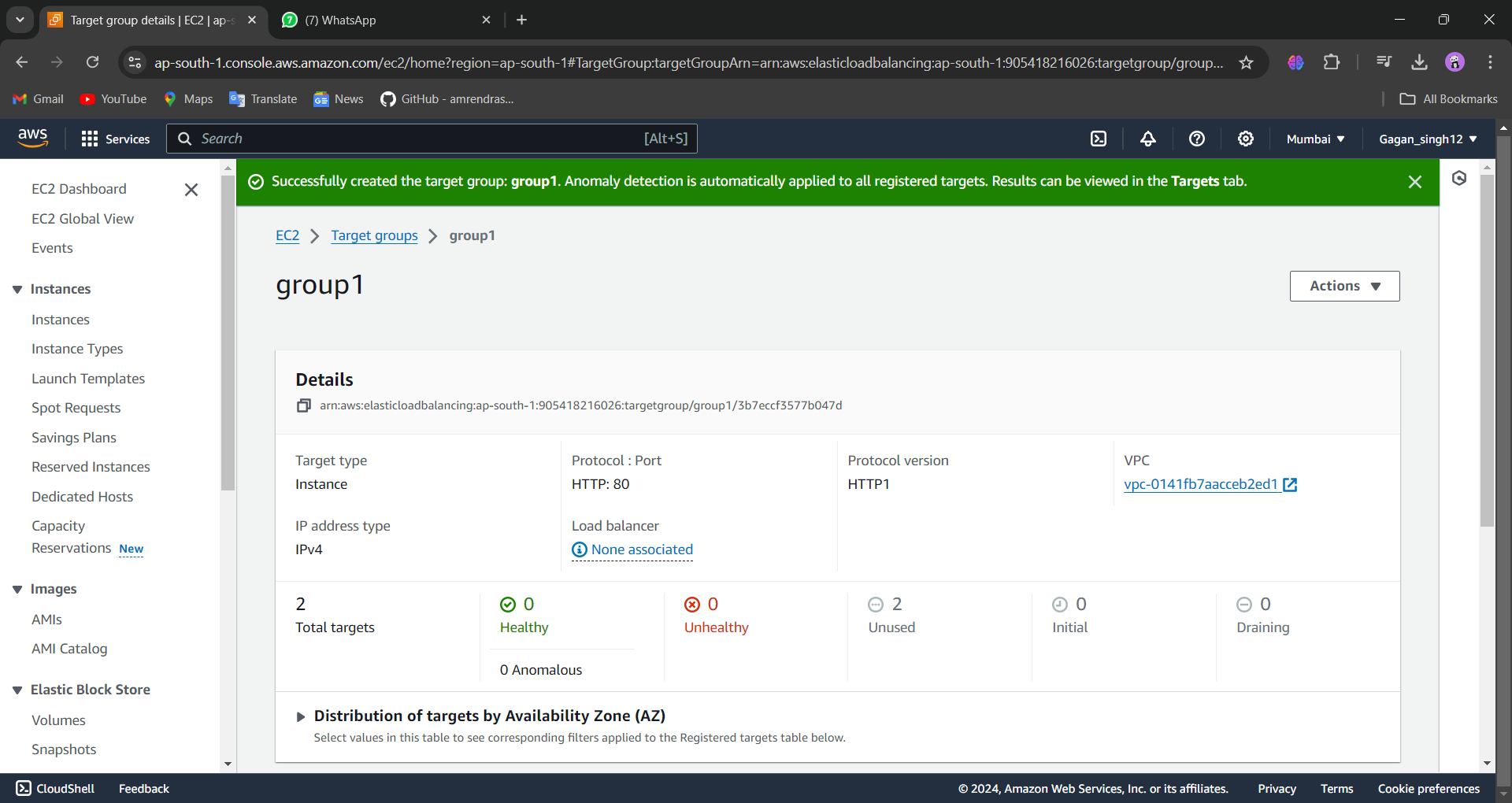
**How to create target group:-**

****

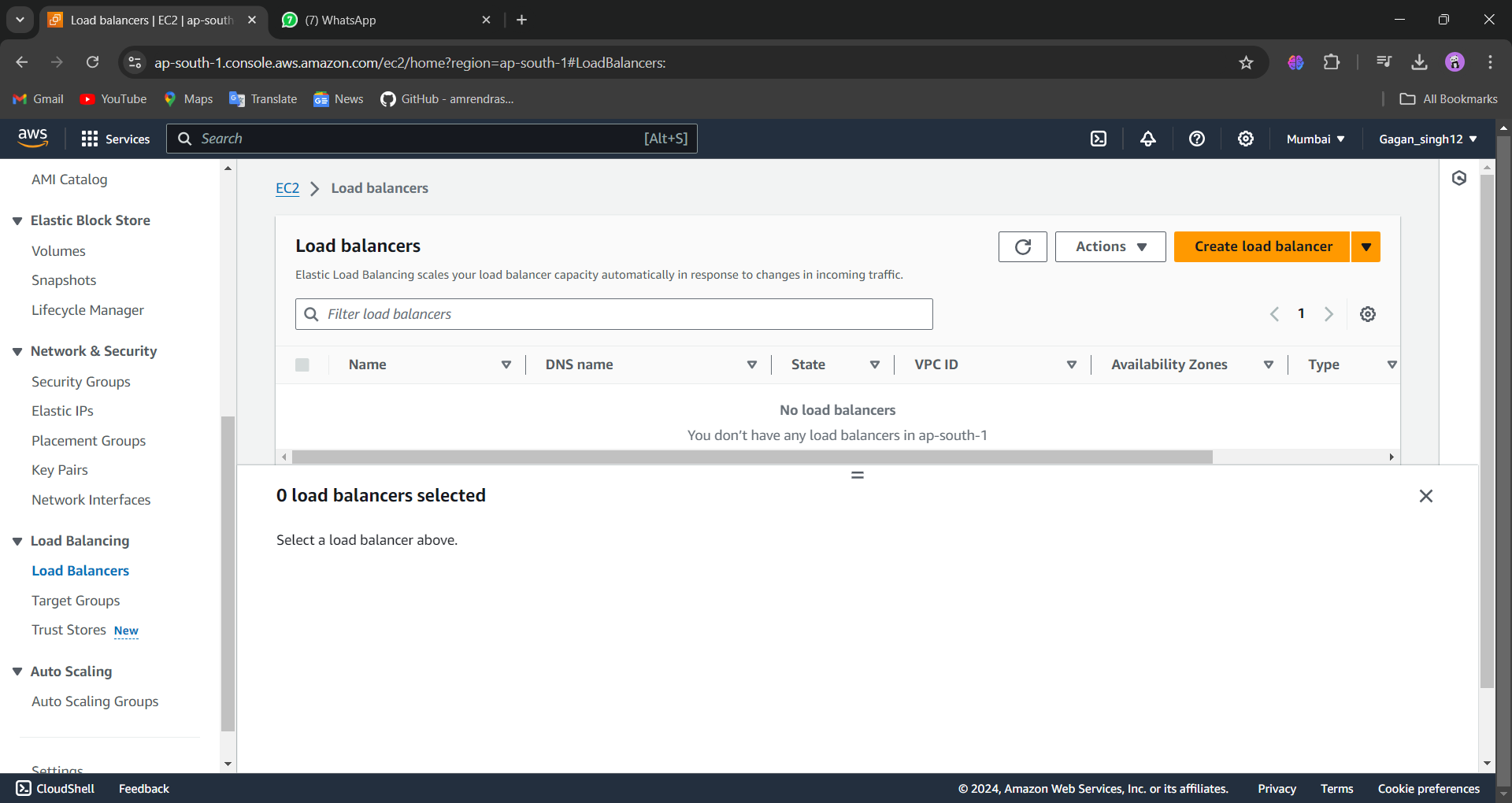
****

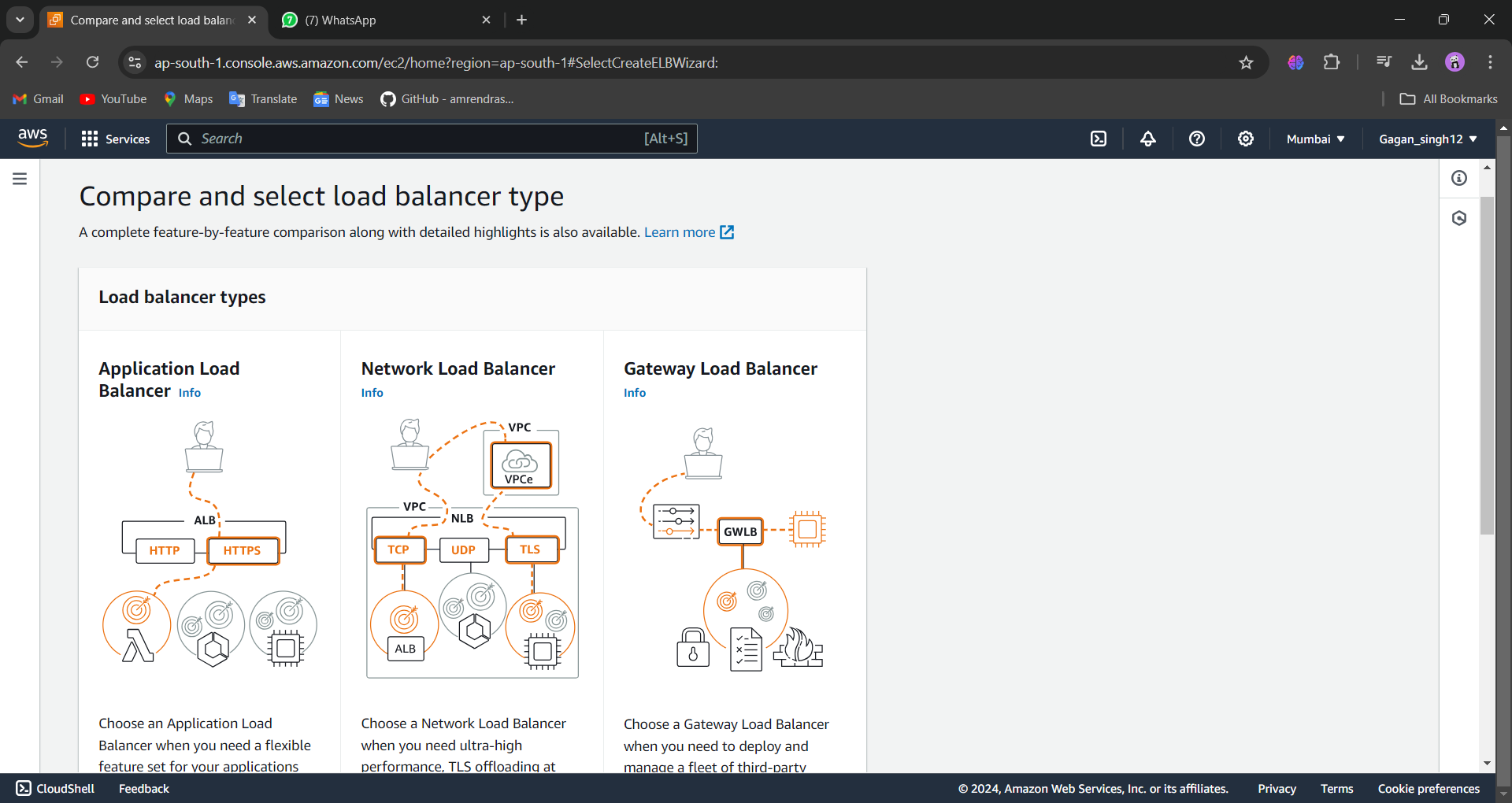
****

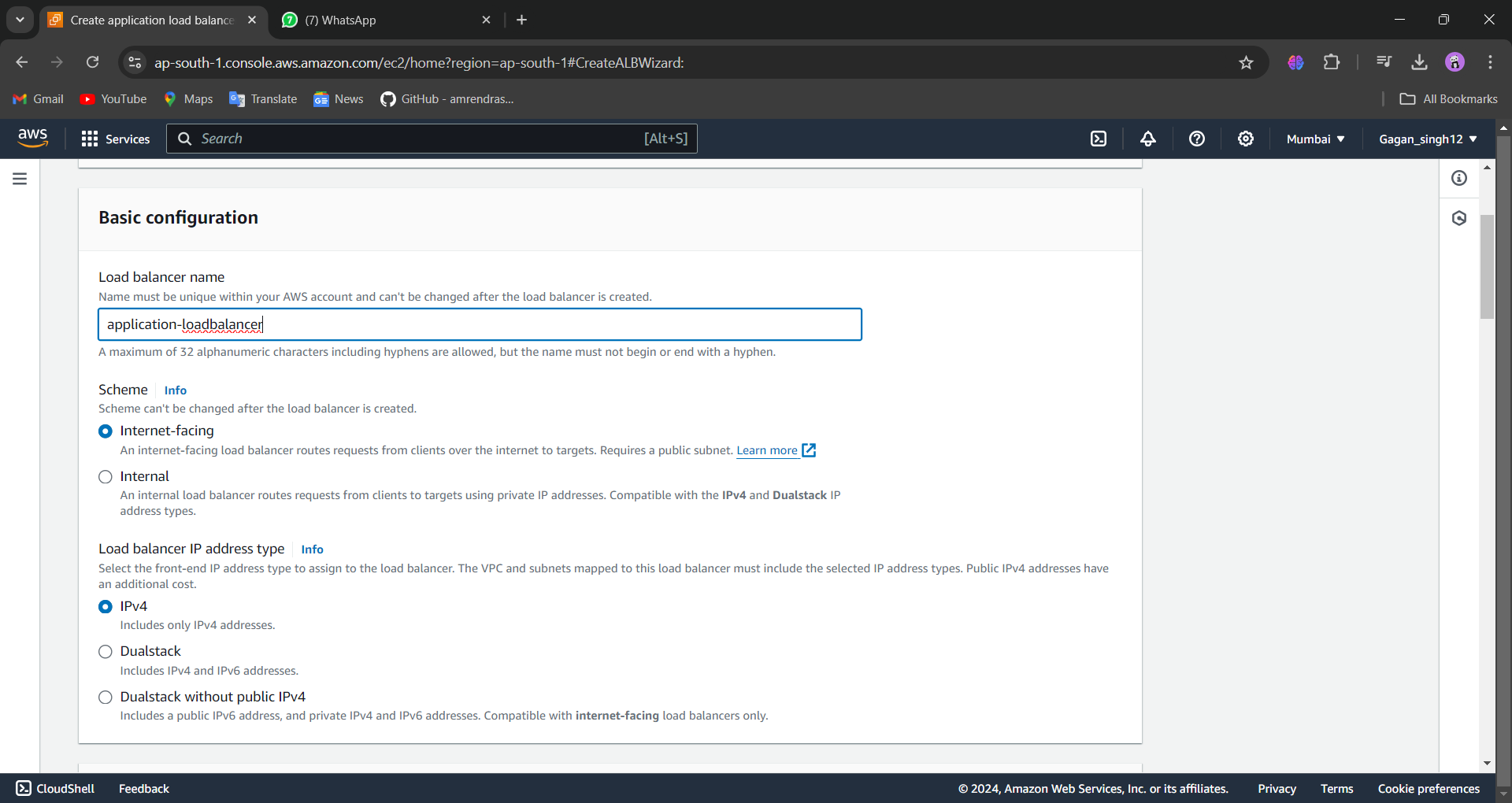
****

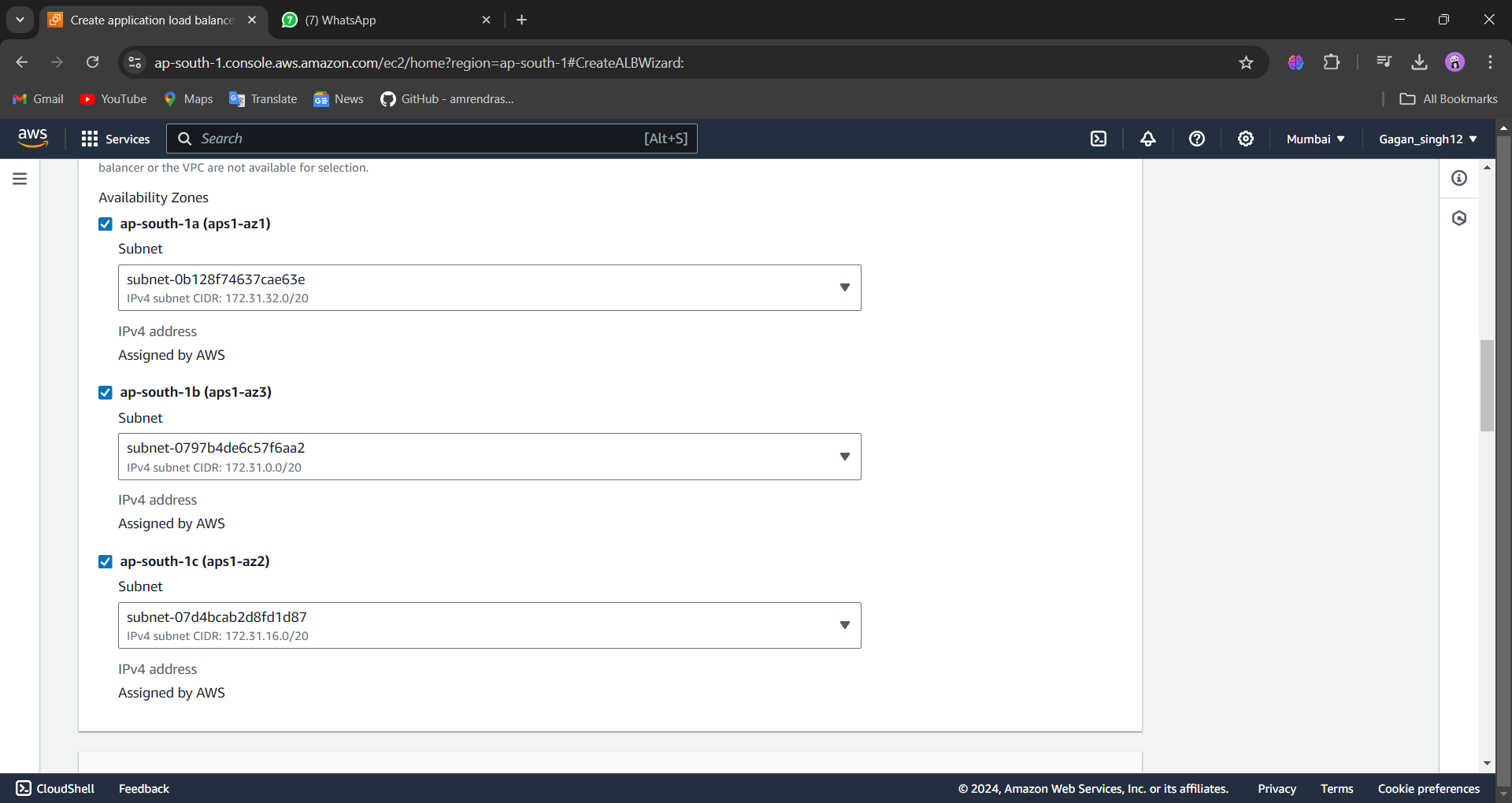
****

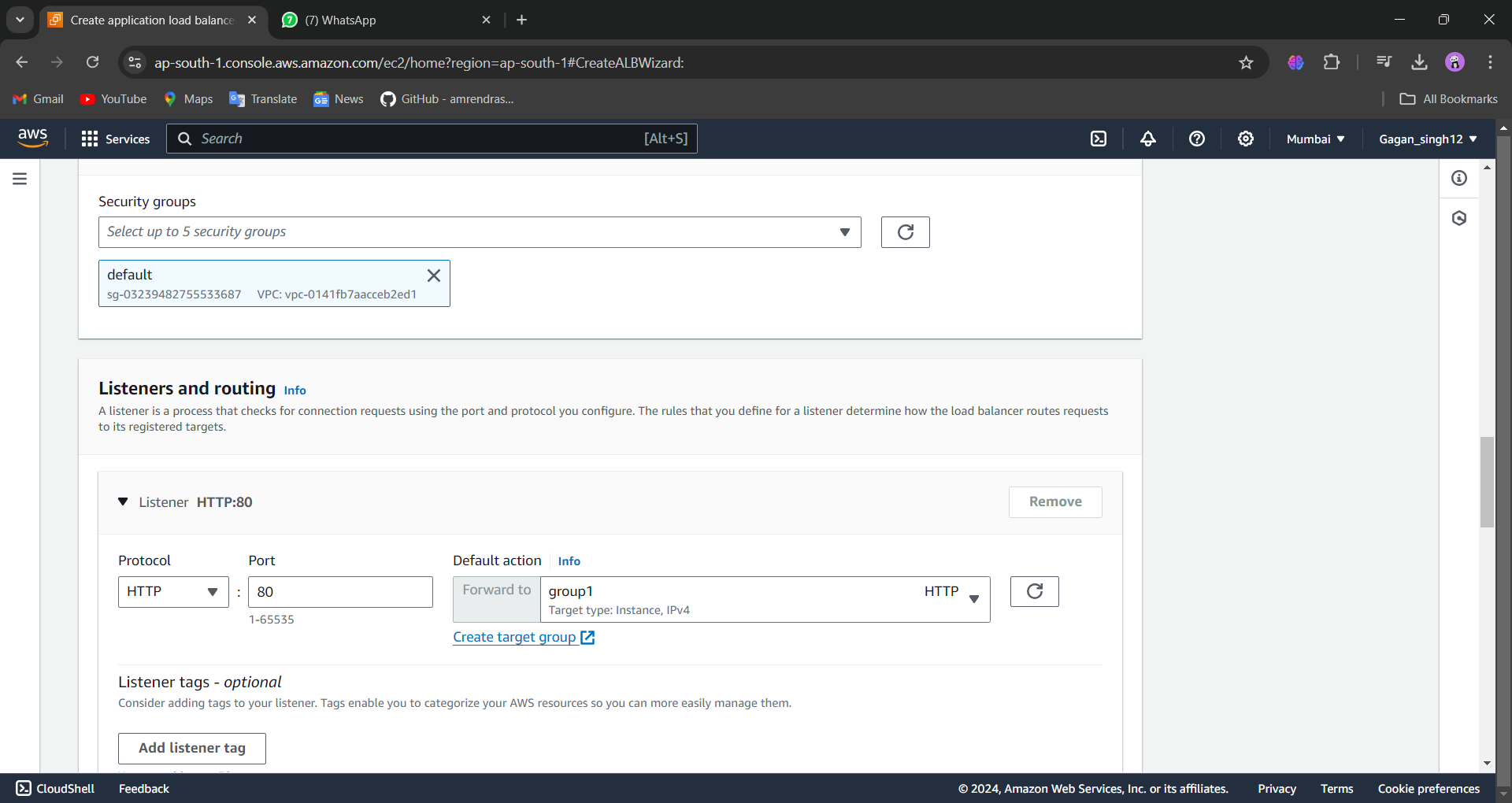
**How to create a load balancer:-**

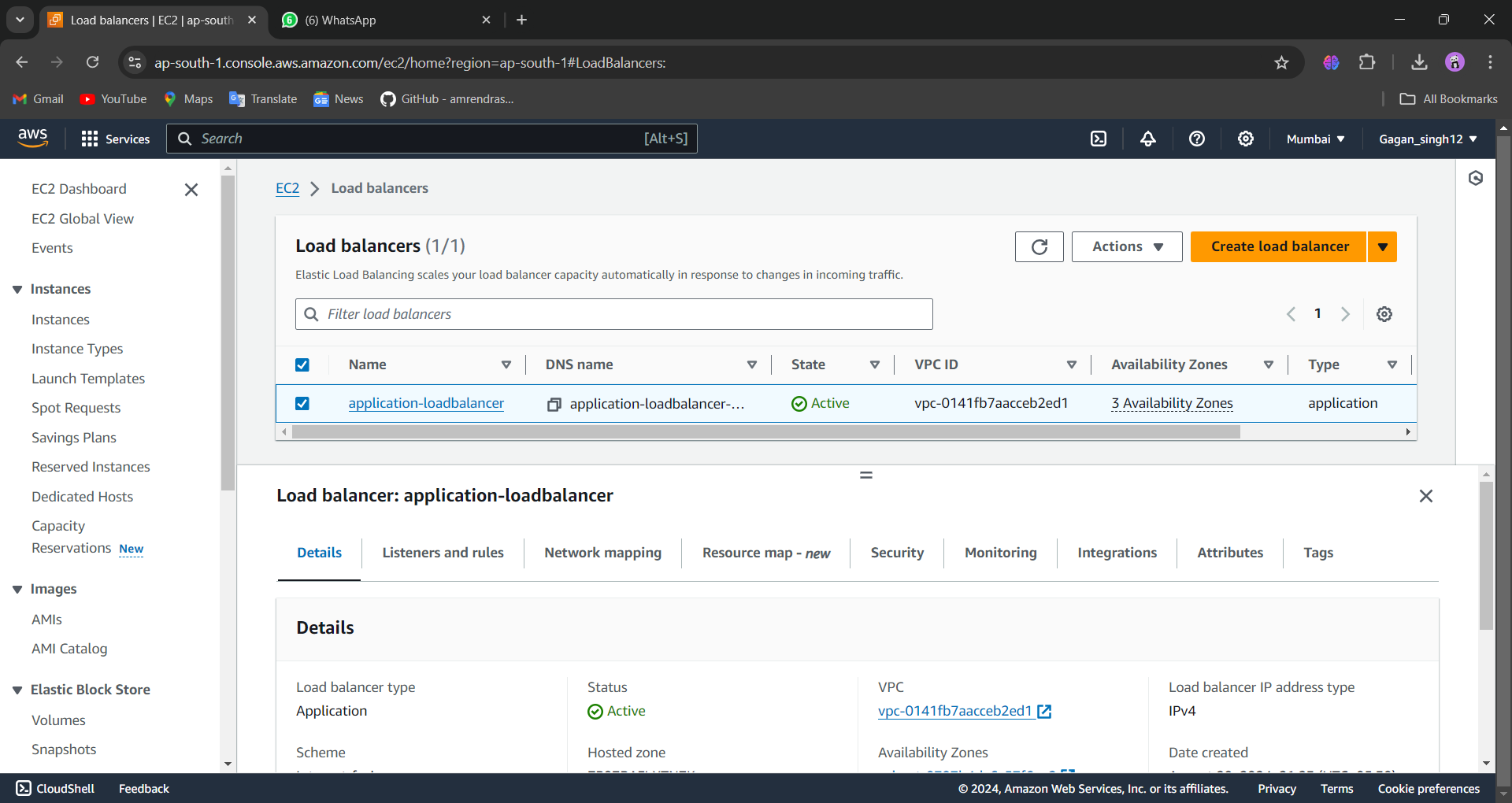
****

****

****

****

****

****

CONCLUSION

Load balancing is a critical concept in network management, particularly for systems that need to handle large volumes of traffic or requests. It involves distributing incoming network traffic across multiple servers to ensure no single server becomes overwhelmed, leading to optimal resource use, reduced latency, and improved reliability.

Here’s a brief explanation with a conclusion:

**Demonstration of Load Balancing**

1. **Scenario**: Imagine a web application that receives requests from users all over the world. Without load balancing, all these requests would go to a single server. If this server gets overloaded, users would experience slow response times or even complete downtime.
2. **Load Balancer in Action**:
   * **Step 1**: A load balancer is introduced between the users and the servers.
   * **Step 2**: The load balancer receives incoming requests and decides which server to forward each request to, based on various algorithms (e.g., Round Robin, Least Connections, IP Hashing).
   * **Step 3**: The servers process the requests and send responses back to the users through the load balancer.
   * **Step 4**: If one server fails or becomes too slow, the load balancer automatically reroutes traffic to the remaining servers, ensuring continuity and efficiency.
3. **Types of Load Balancing Algorithms**:
   * **Round Robin**: Distributes requests sequentially across the servers.
   * **Least Connections**: Directs traffic to the server with the fewest active connections.
   * **IP Hashing**: Routes requests based on the client’s IP address.

**Conclusion**

Load balancing plays a vital role in maintaining the performance and availability of applications. By efficiently distributing incoming traffic among multiple servers, it prevents any single server from being overwhelmed, ensuring that applications remain responsive and reliable even under heavy load. This not only enhances user experience but also contributes to the scalability and resilience of the system.