**PRACTICAL: 5**

**AIM:**

Implementing Load balancing/High Availability for Web Application using Amazon Web Services (AWS):- Classic Load Balancing, Network load balancing & Application Load balancing.

**THEORY:**

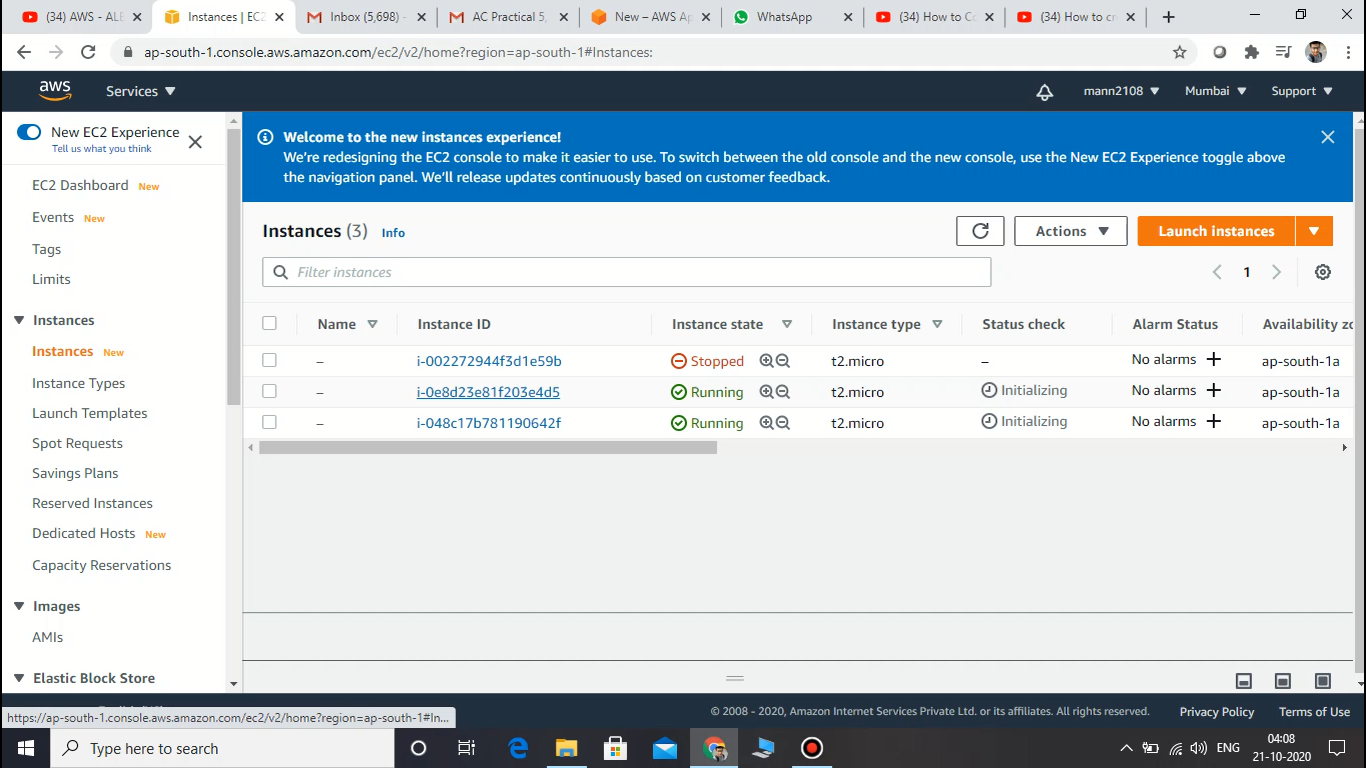


**Classic Load Balancer** provides basic load balancing across multiple Amazon EC2 instances and operates at both the request level and connection level. Classic Load Balancer is intended for applications that were built within the EC2-Classic network.

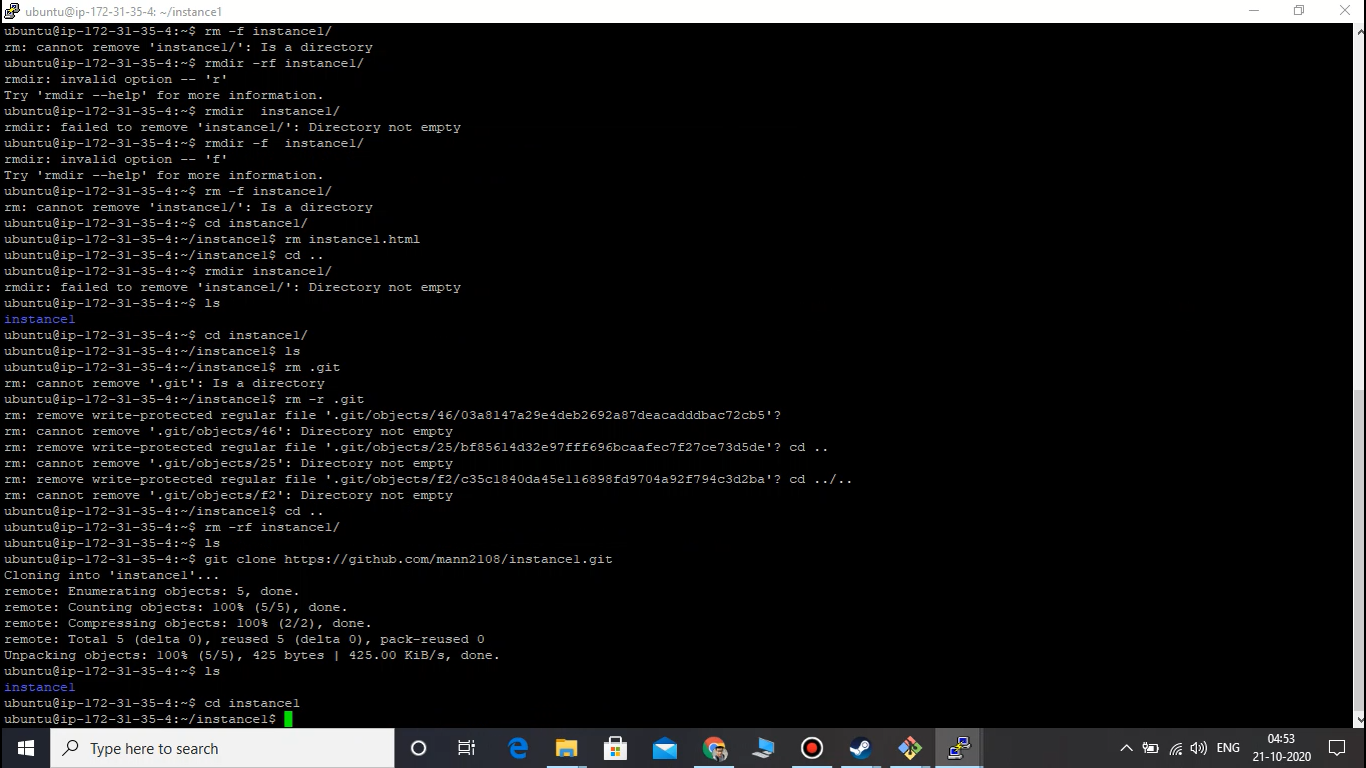
**Network Load Balancer** can be used as a single point of contact for the clients. This balancer distributes the incoming requests to multiple targets, and one such target is the Amazon EC2 instance. When requests are spread out, the availability of the service improves.

**An Application Load Balancer** functions at the application layer, 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.

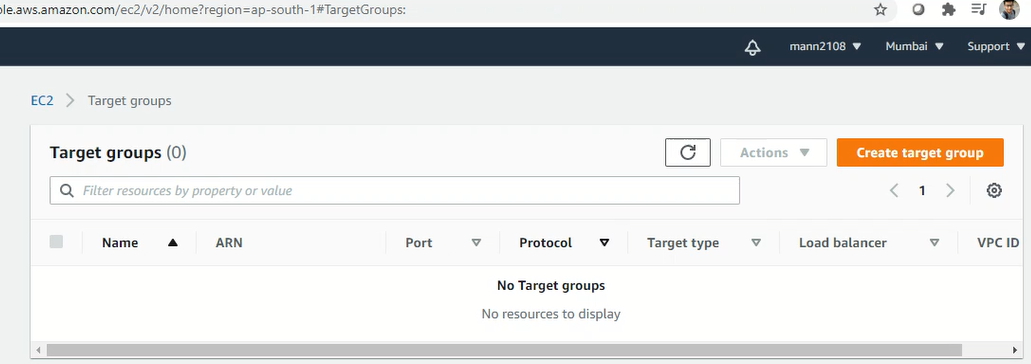
**OUTPUT: (Creating Application Load Balancer)**



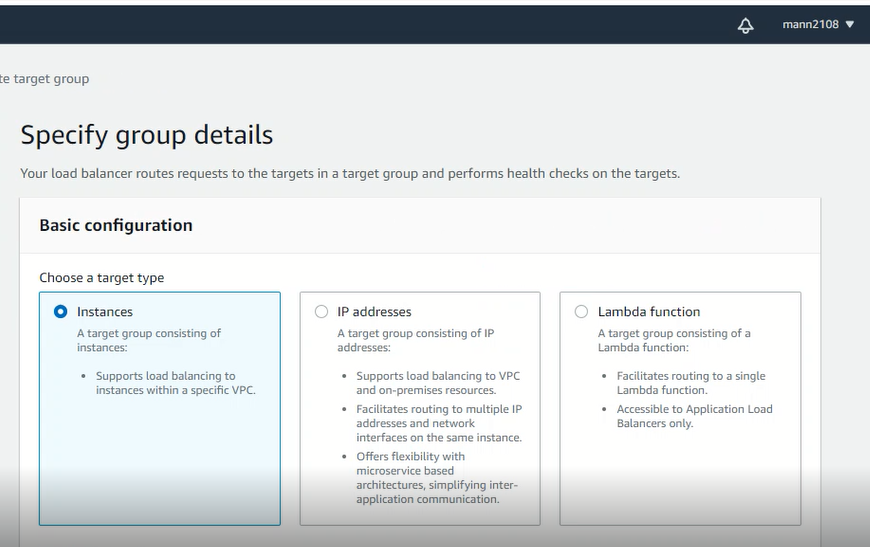
**Creating two ec2 instances under free tier configurations**



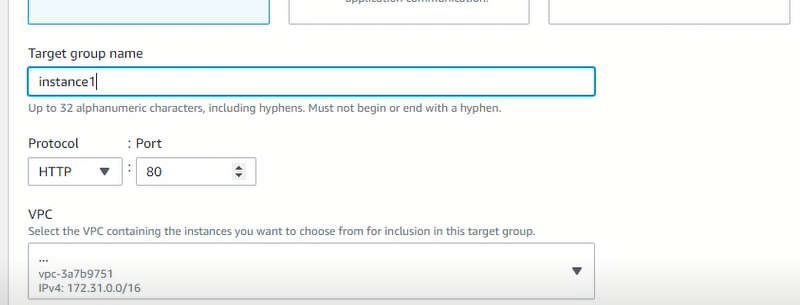
**Using PUTTY to connect to both ec2 instance from windows using private keys with “.pem” format**



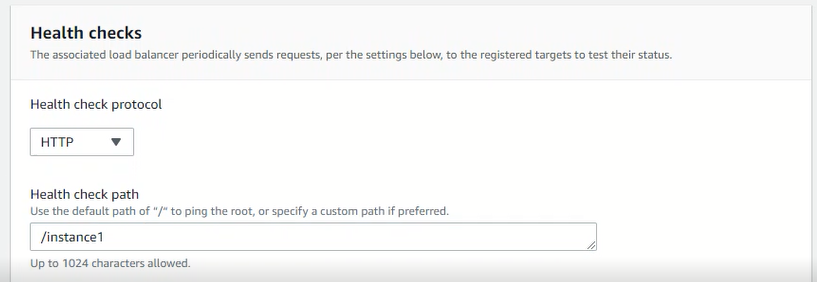
**Creating target groups**



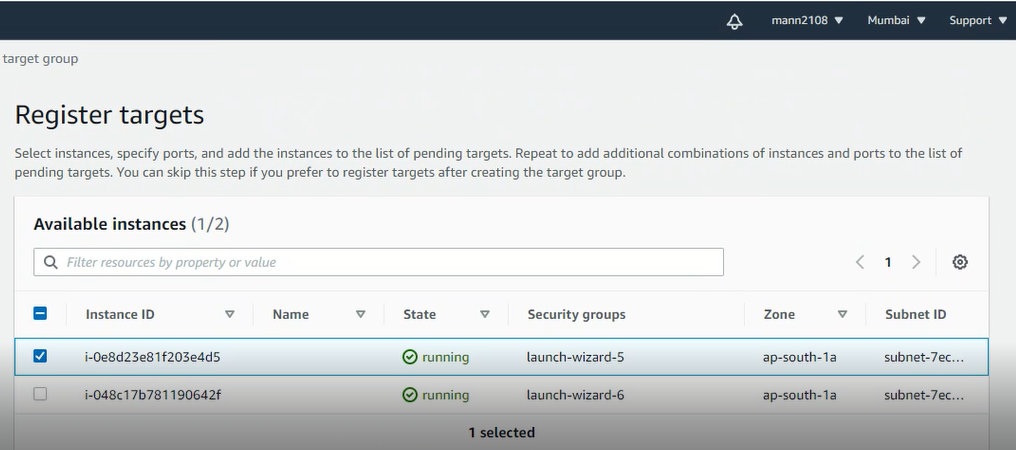
**Specify group details with Instances**



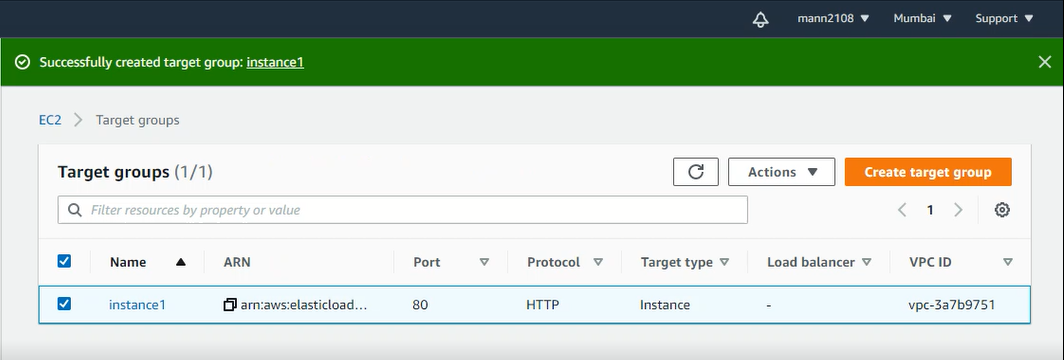
**Giving name to the group**



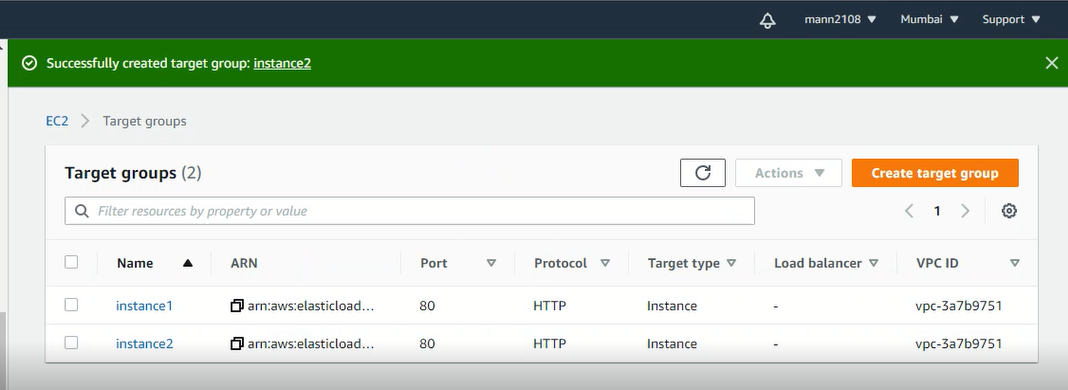
**Providing health checks**



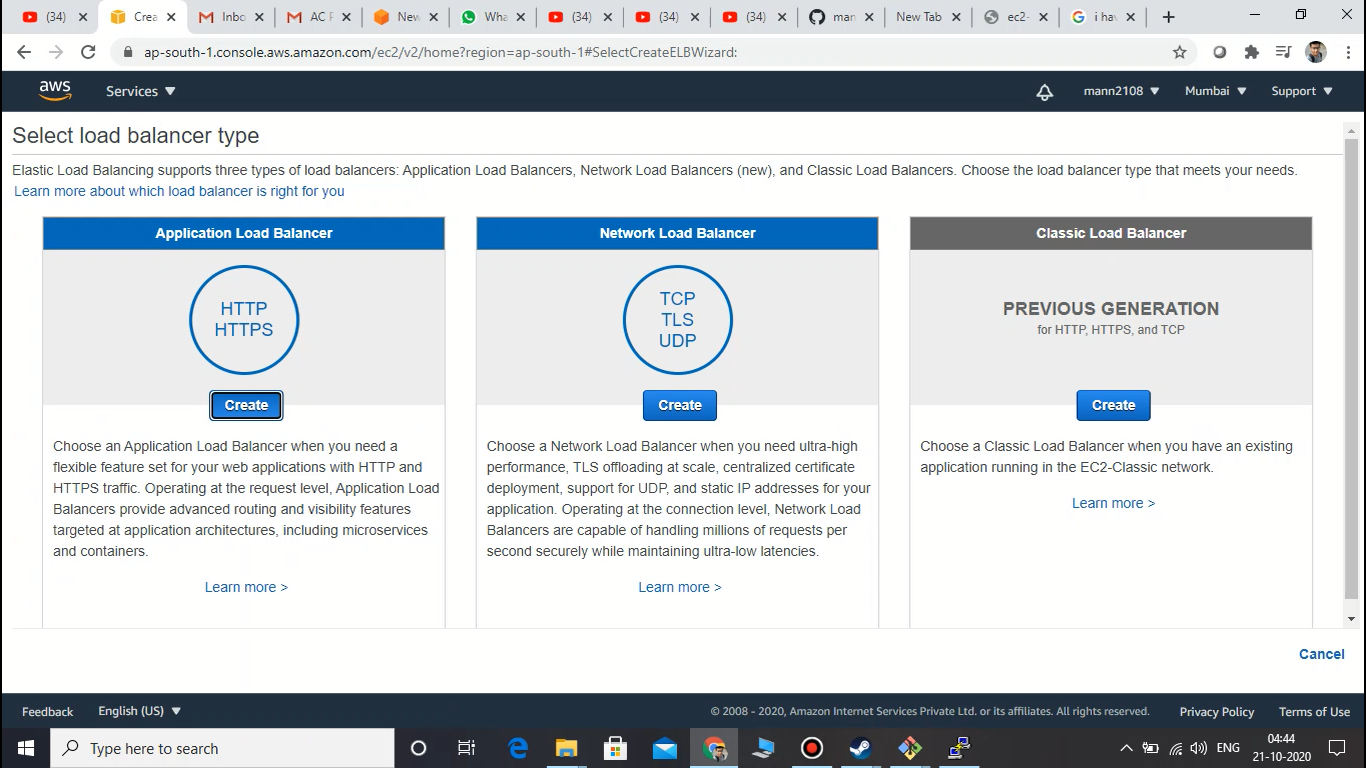
**Register with one of the instances**



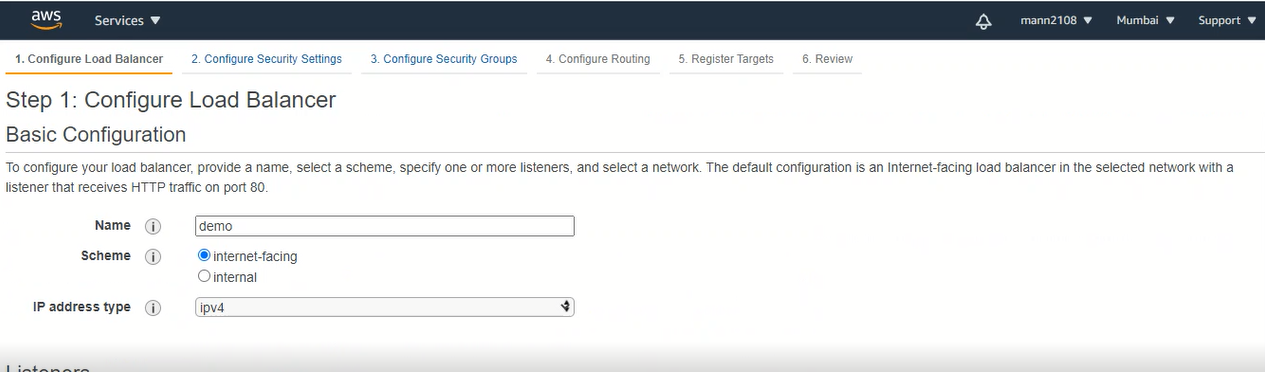
**First target group created successfully**



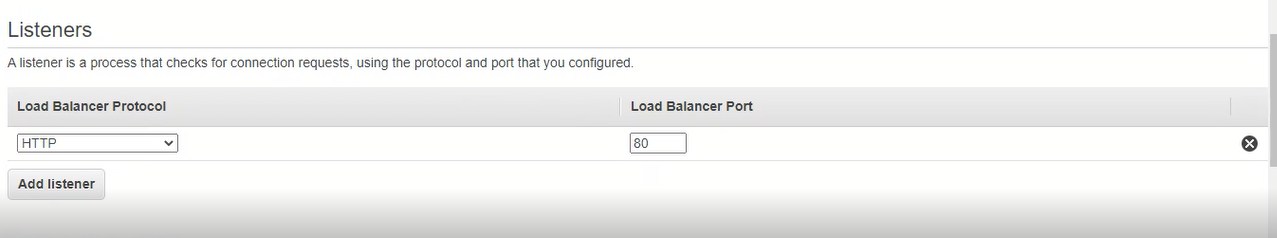
**Similarly created another target group with another instance as a target**



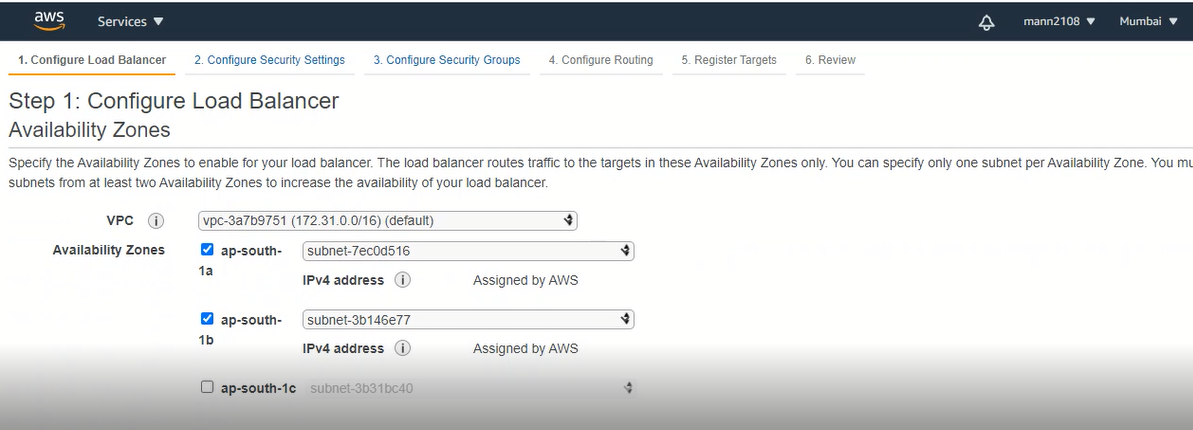
**Creatign HTTP/HTTPs load balancer**



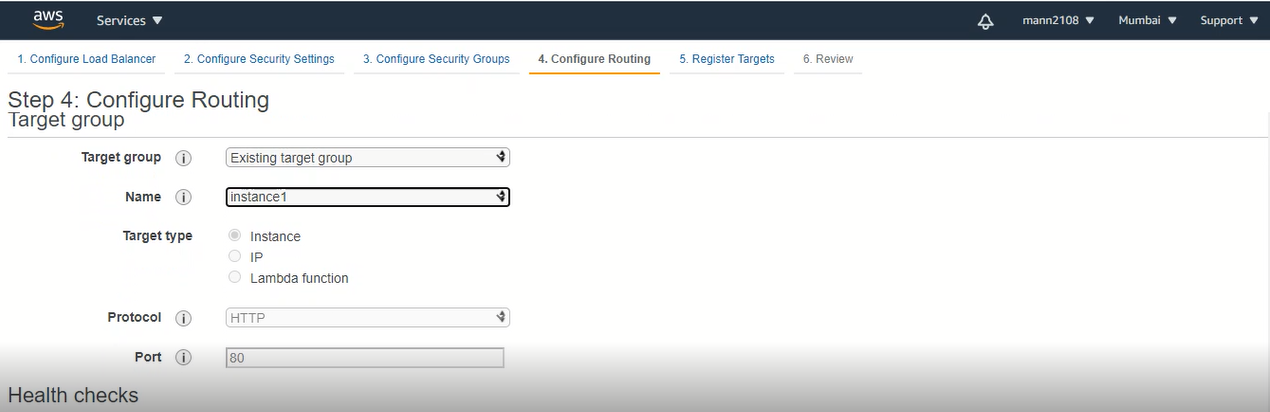
**Providing name as demo to load balancer**



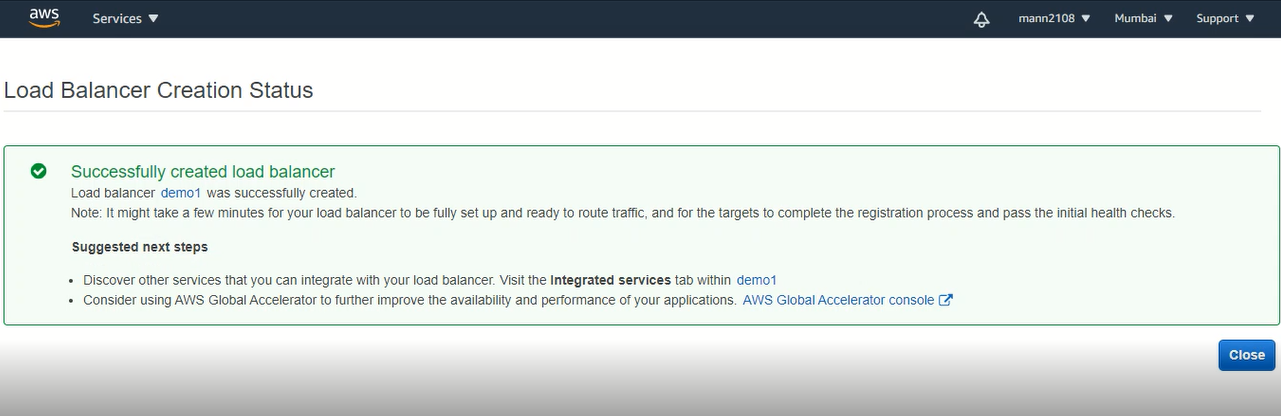
**Assigning protocol and port**



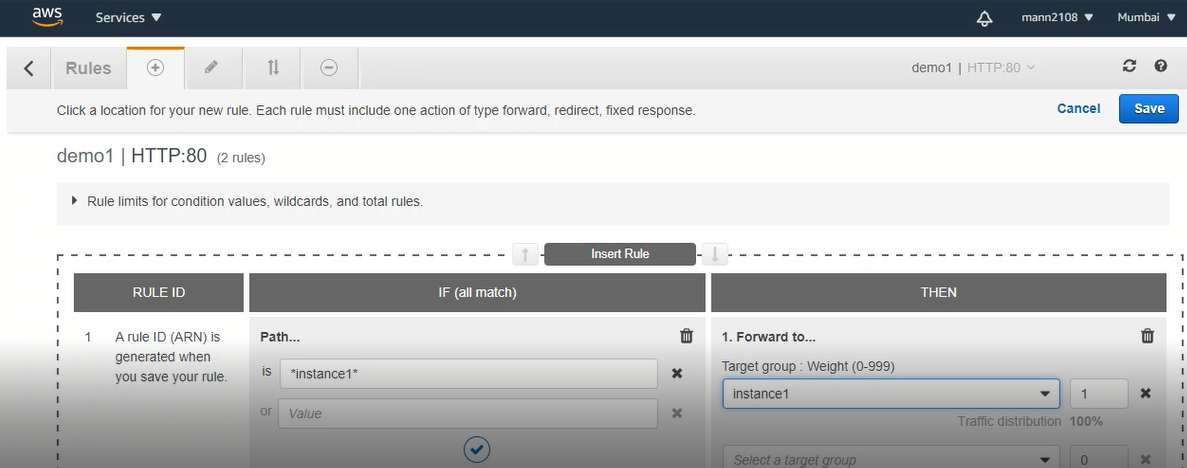
**Configure load balancer by assigning two subnets atleast 2 are mandatory**



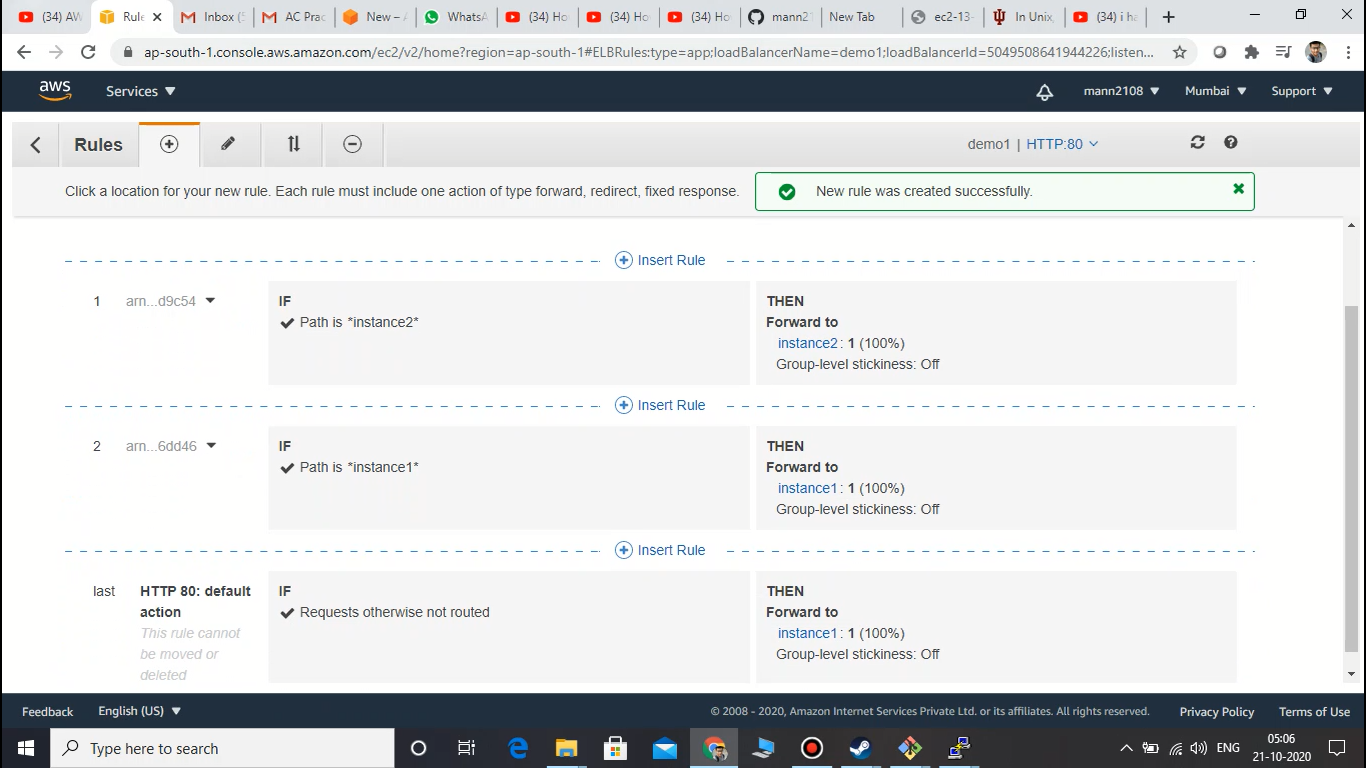
**Selecting existing target group instance 1**



**Successfully created load balancer**



**Editing listners**



**Assigning forwarding on the basis of target paths**

**LATEST APPLICATIONS:**

Load balancing is the process of distributing workloads across multiple servers, collectively known as a server cluster. The main purpose of load balancing is to prevent any single server from getting overloaded and possibly breaking down. In other words, load balancing improves service availability and helps prevent downtimes.

Also, when the amount of workload an individual server receives is within its acceptable levels, it would in turn have sufficient computing resources (e.g. CPU, RAM) to process requests within acceptable response times. Fast response times are vital to end user satisfaction and productivity.

**LEARNING OUTCOME:**

In this lab we first started with creating two ec2 instances from the aws console under free tier configurations, then we connect both instance from windows local machine using PUTTY and private key generated during creating instances by default for single machine we use single key pair for each instances, then we create two target groups with two instances respectively after that we move towards creating a HTTP/HTTPs load balancer in which we assign both the target groups and forwarding path rules hence we implemented the load balancing successfully.

**REFERENCE:**

NLB :- Qwiklab :- Introduction to Elastic Load Balancing

ALB:- <https://aws.amazon.com/blogs/aws/new-aws-application-load-balancer/>