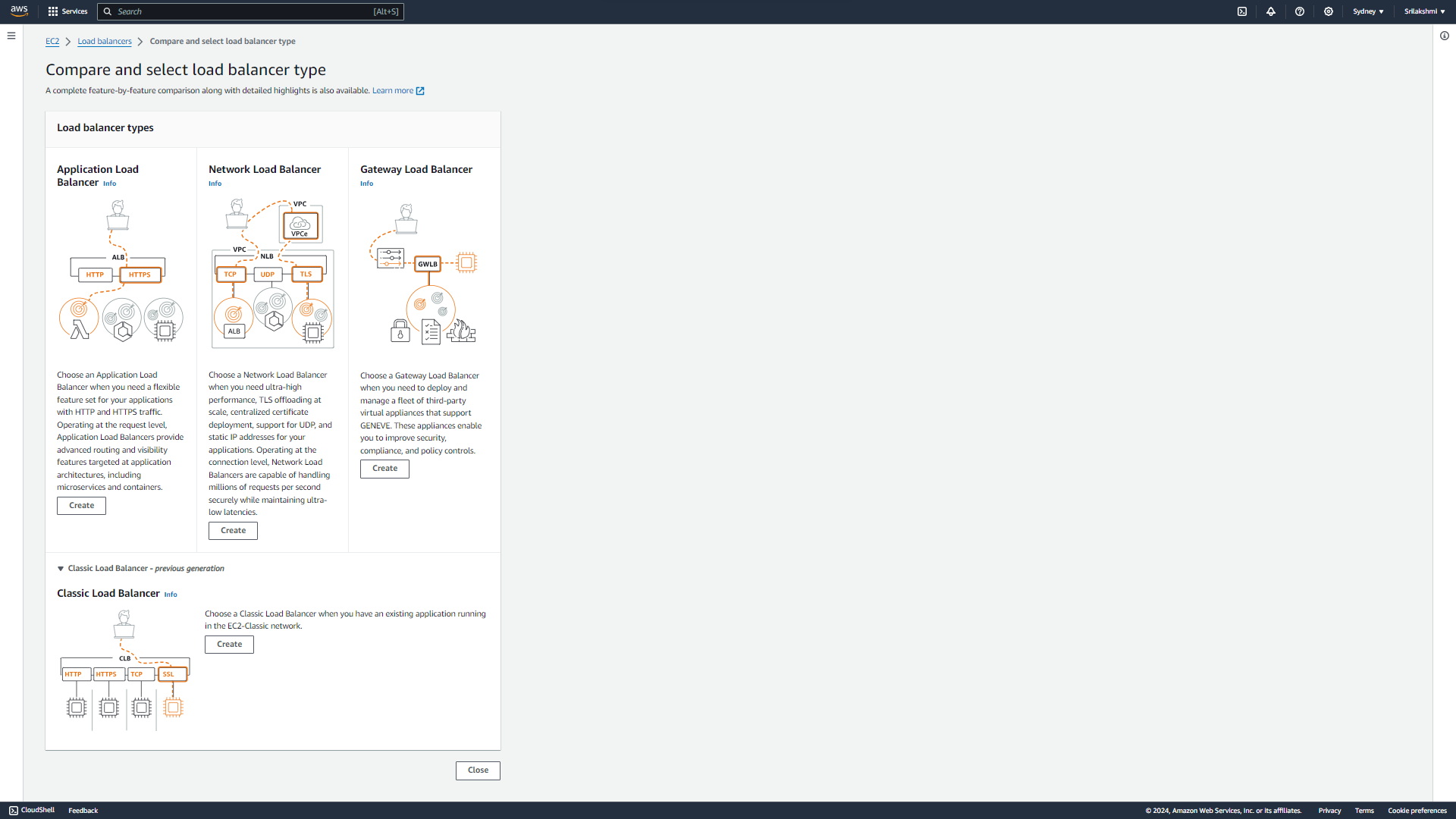
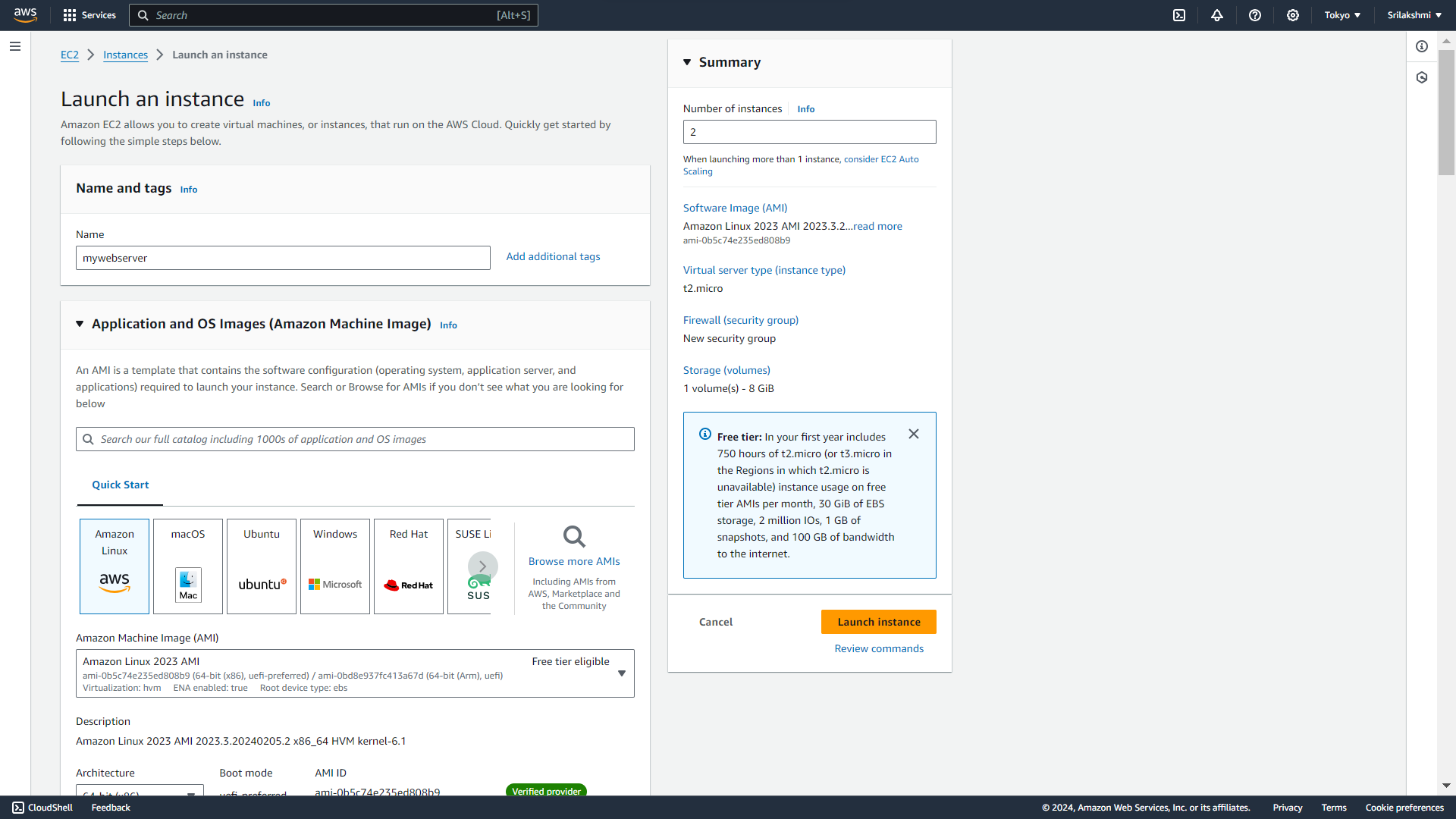
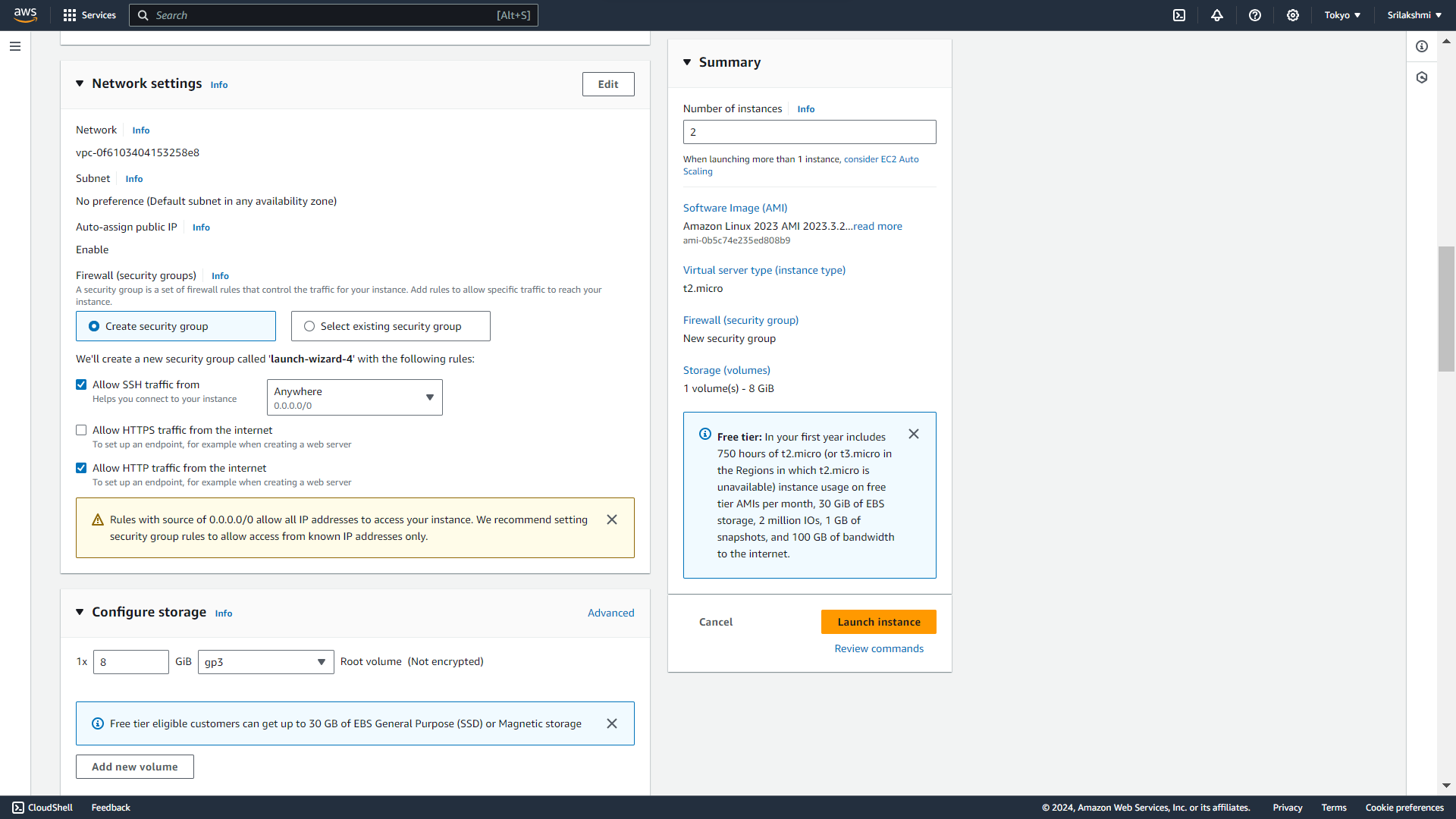
Implement load balancing and high availability to open-source application  
**Part1:** Explore the methodologies and possibilities that can be implemented in load balancing and creating an application with high availability.

**Part2:** Implement one of the explored methodologies in real time.

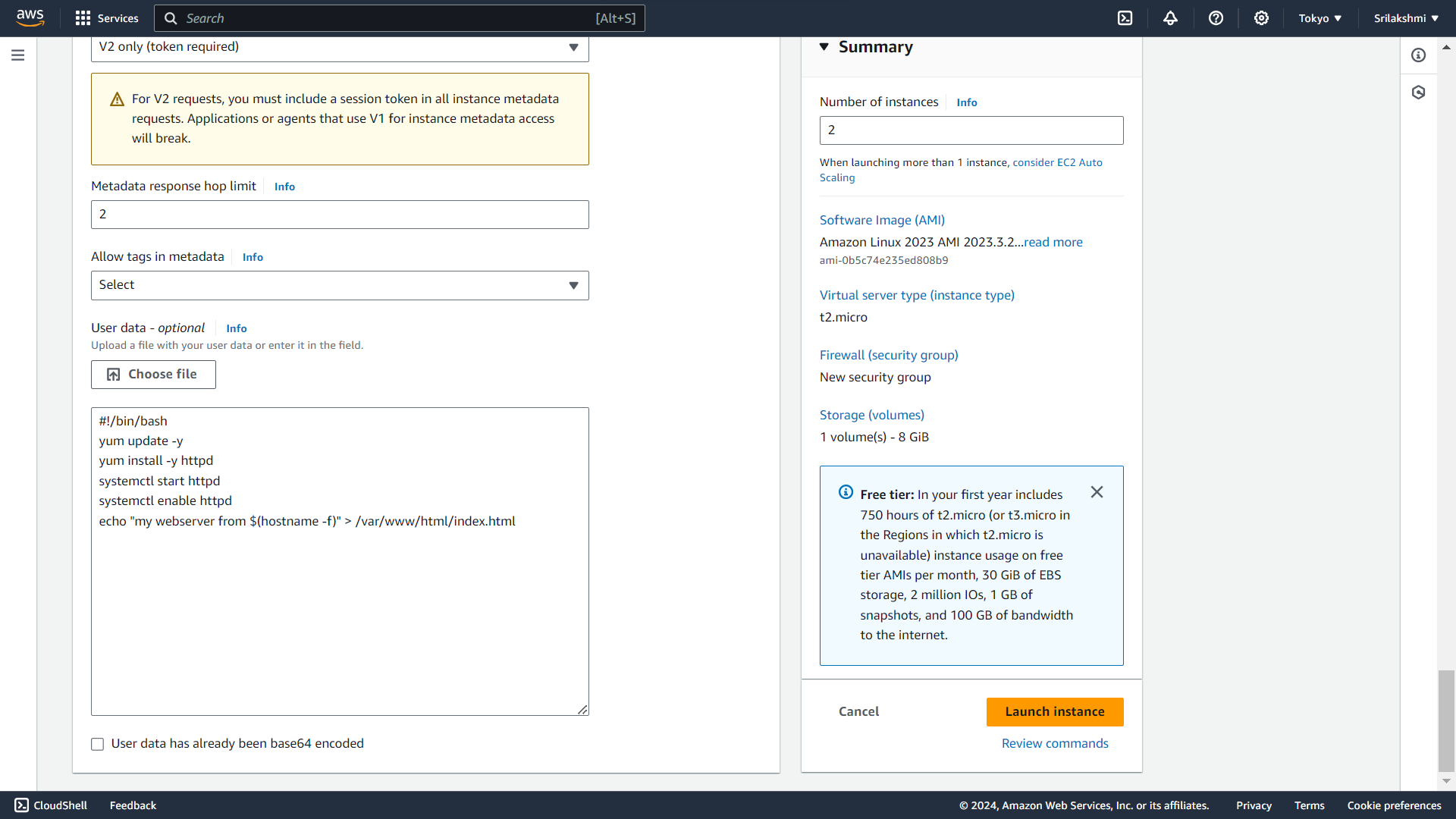
**  
Load Balancer:** Load balancer is a type of server that distributes the incoming web traffic across multiple backend servers.  
**Advantages:**1. Improves performance   
2. Reduce response time   
3. Increase high availability **There are four types of load balancers**   
1. Classic load balancer  
2. Application load balancer  
3. Network load balancer  
4. Gate way load balancer

**1.Classic Load Balancer:** It is the traditional form of load balancer which was used initially. It distributes the traffic among the instances directly and it is not intelligent enough to support host-based routing or path-based routing. It ends up reducing efficiency and performance in certain situations.  
Classic load balancer works in transport layer (TCP/UDP) or the application layer (HTTP/HTTPS). It is used in EC2 classic servers (no need of VPC) It is cost effective.   
**2**. **Application Load Balancer:** This load balancer works on the application layer of the OSI model. This type of load balancer is used when decisions are to be made related to HTTP and HTTPS traffic routing. It supports path-based routing and host-based routing. The load balancer also supports dynamic host port mapping**.  
3. Network Load Balancer:** This type of works on the transport layer of (TCP and UDP protocols). This Load balancer will configure only based on Ip’s or ports. It is capable of handling millions of requests per second. It is mainly used for load balancing TCP traffic**.** This load balancer provides static IP.  
**4**. **Gateway Load Balancer**: Gateway Load balancer is used for security purpose. It works on transport layer and network layer. It provides you the facility to deploy, scale, and manage virtual appliances like firewall. Gateway Load balancers combines a transparent network gateway and then distributes the traffic.   
**Part-2: Implementing Application load balancer in real time**  
**Login to Aws console and Create 2 EC2 instances-my webserver  
OS-Amazon Linux**

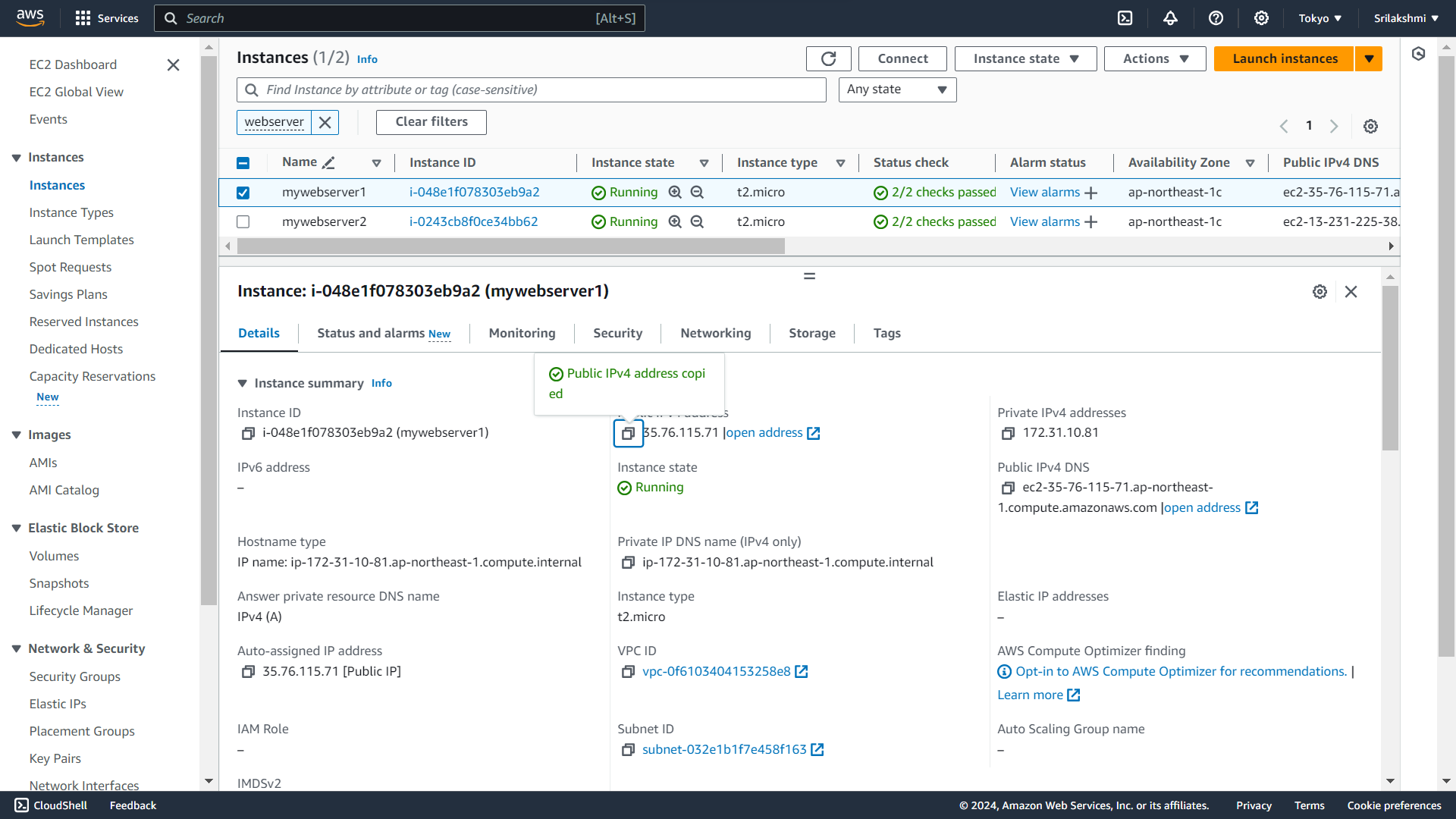
**Select default network settings**

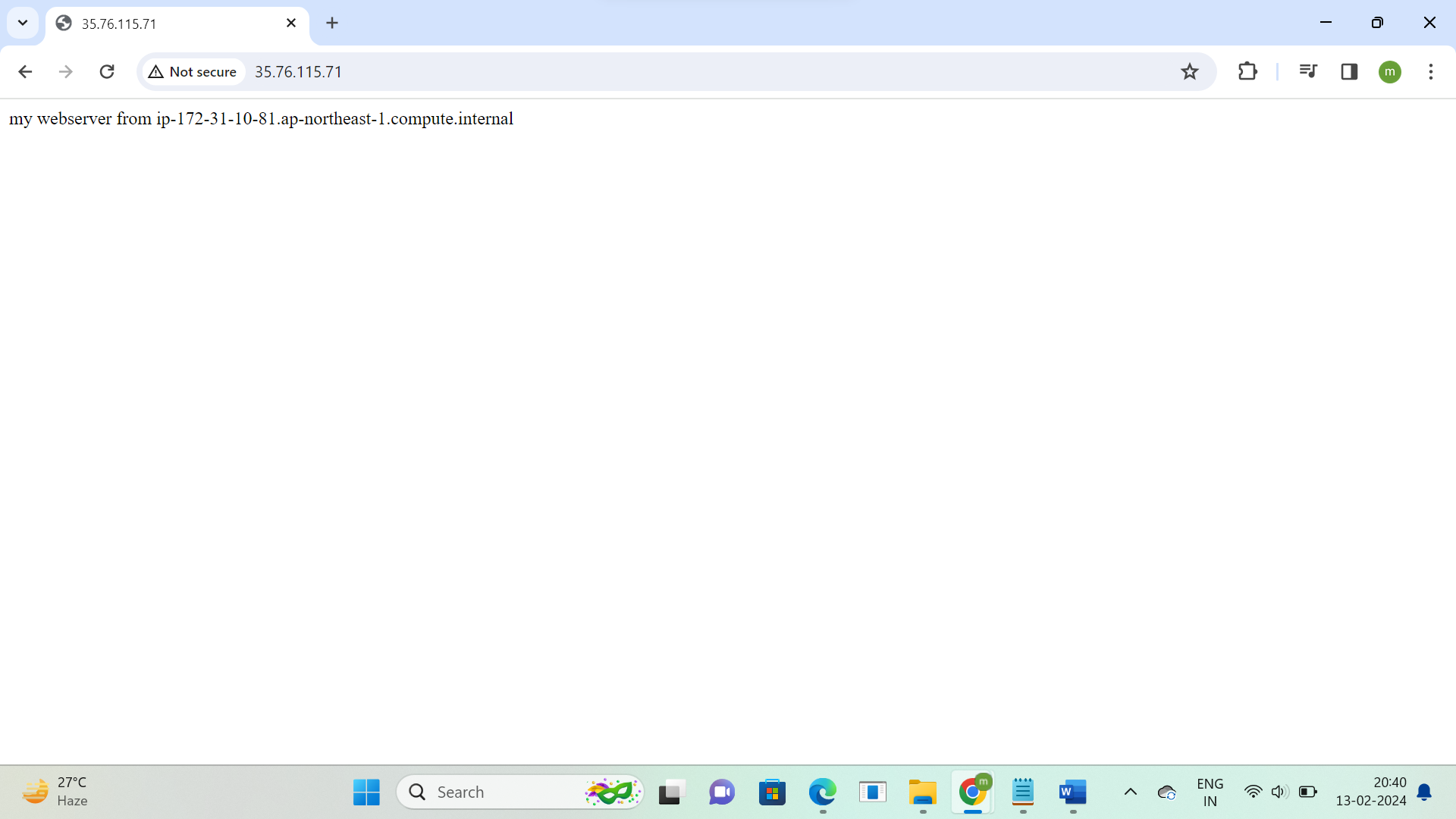


**Enter this script and click on launch instance**

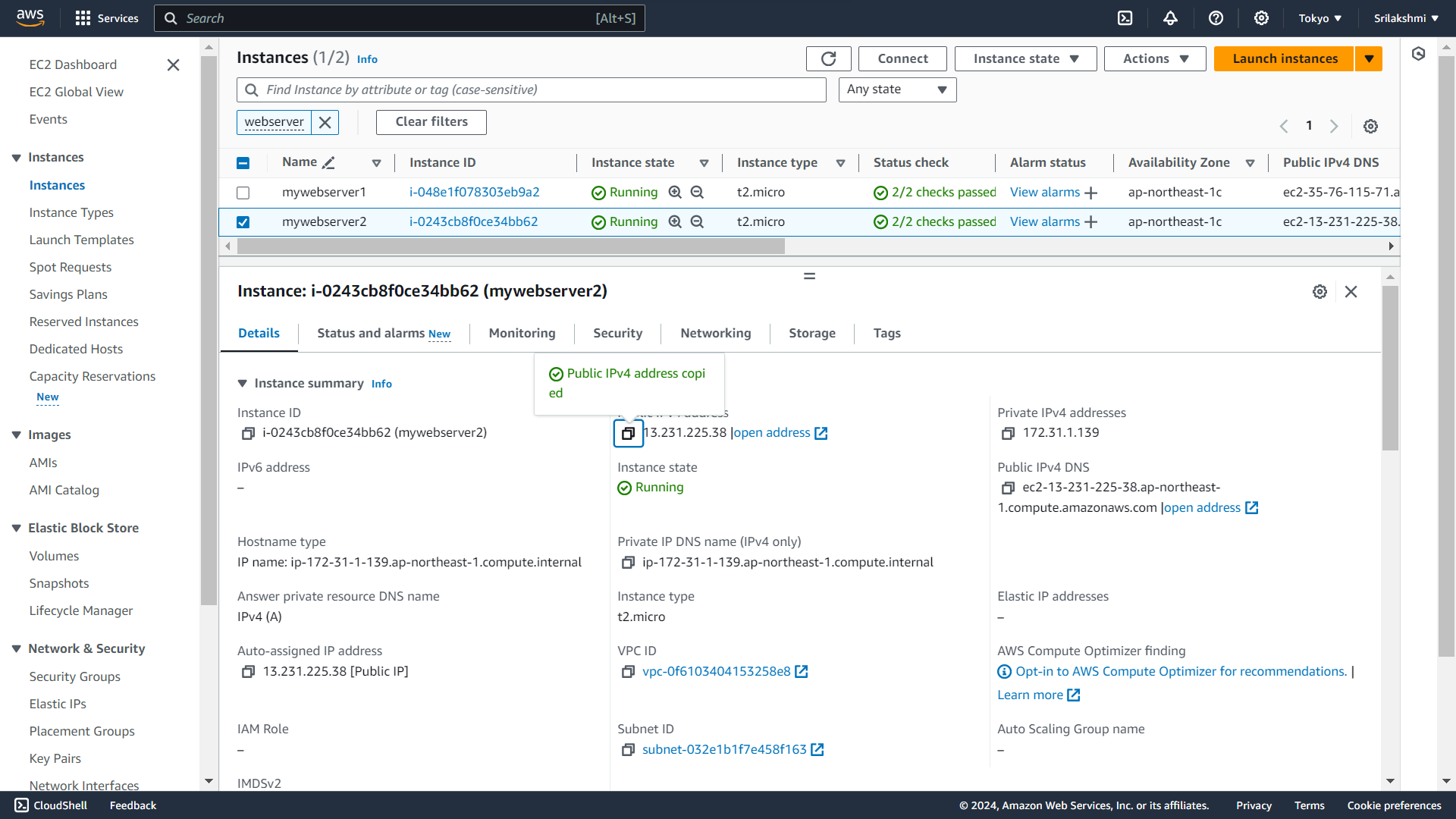
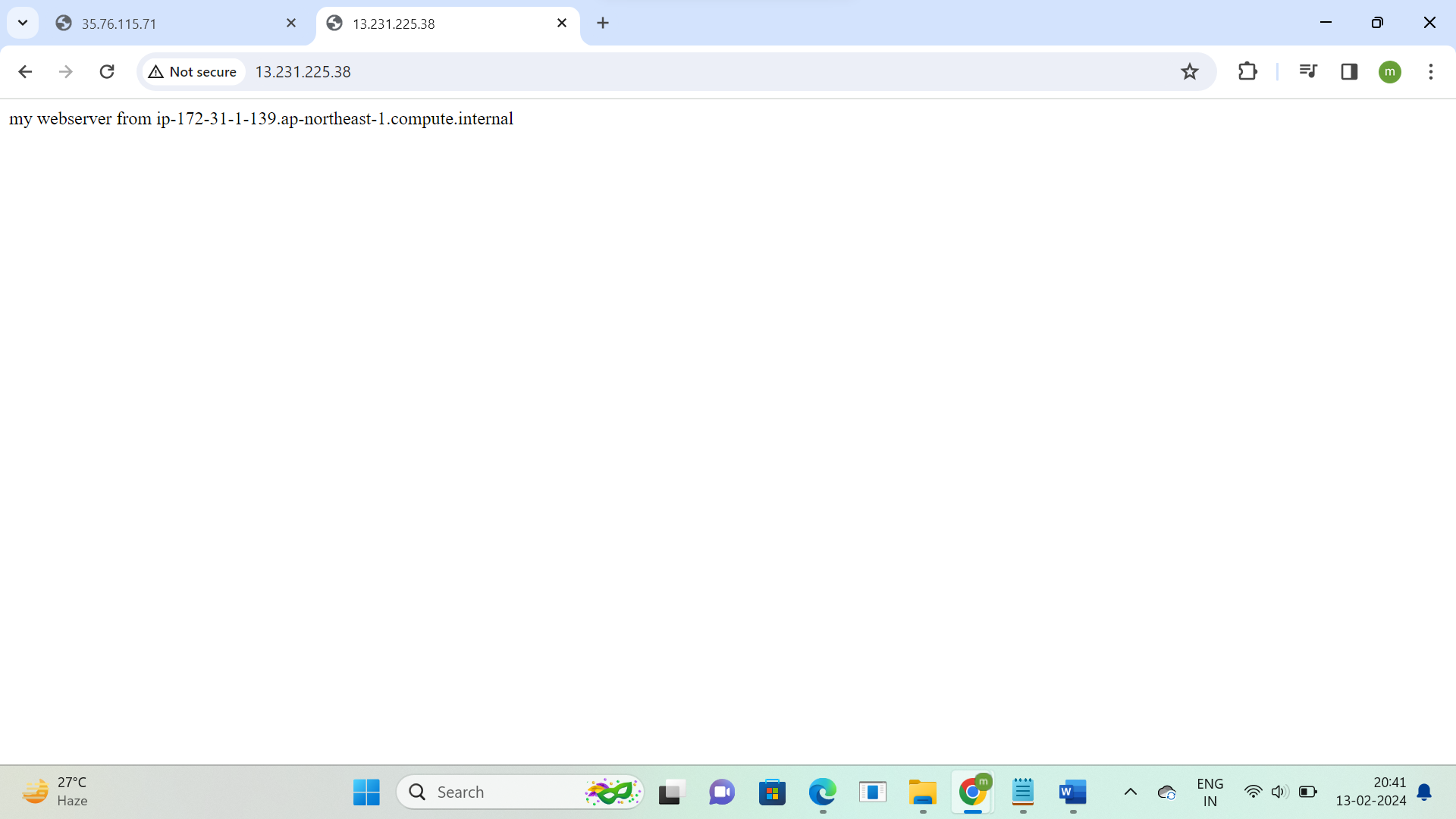
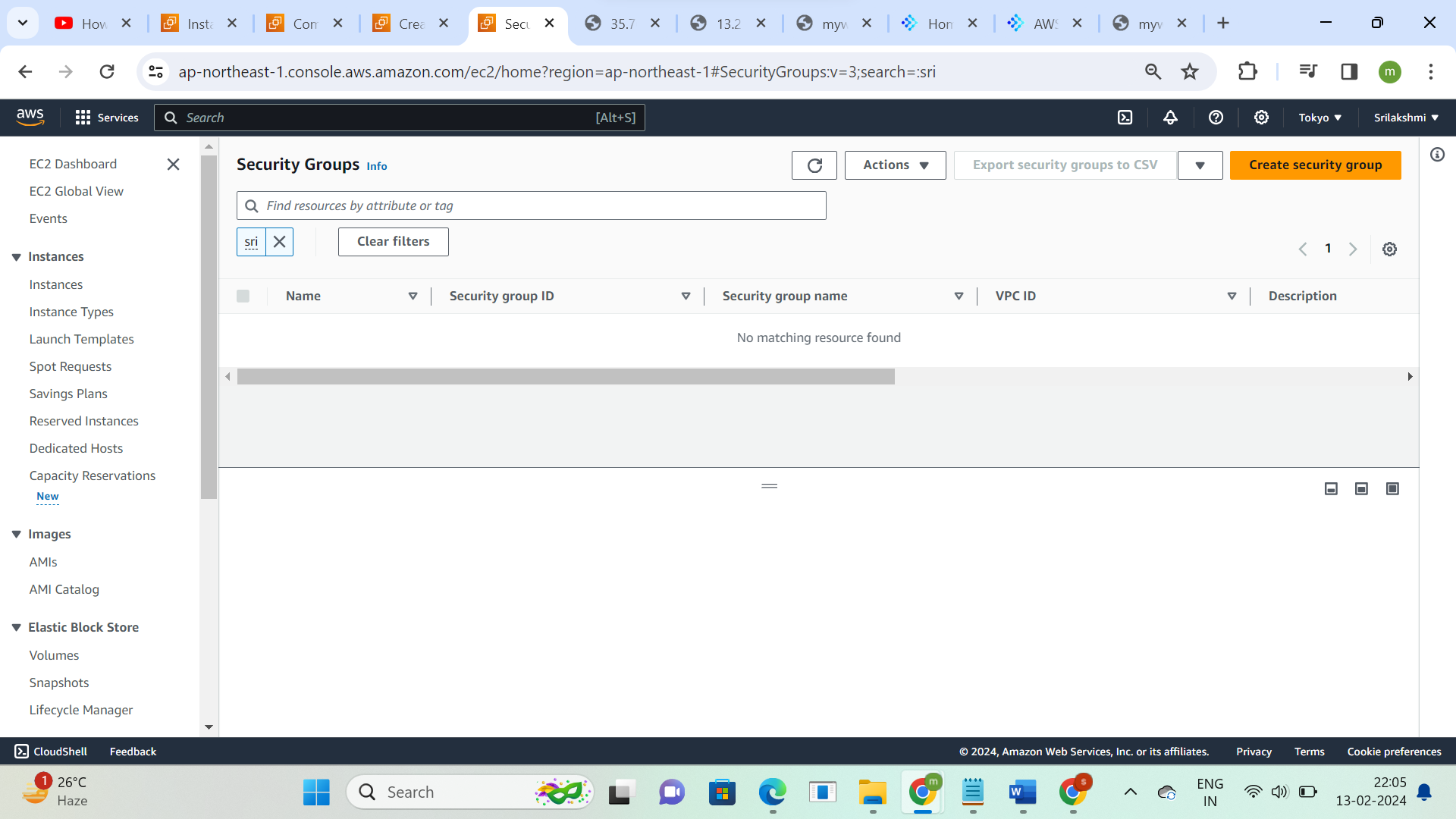
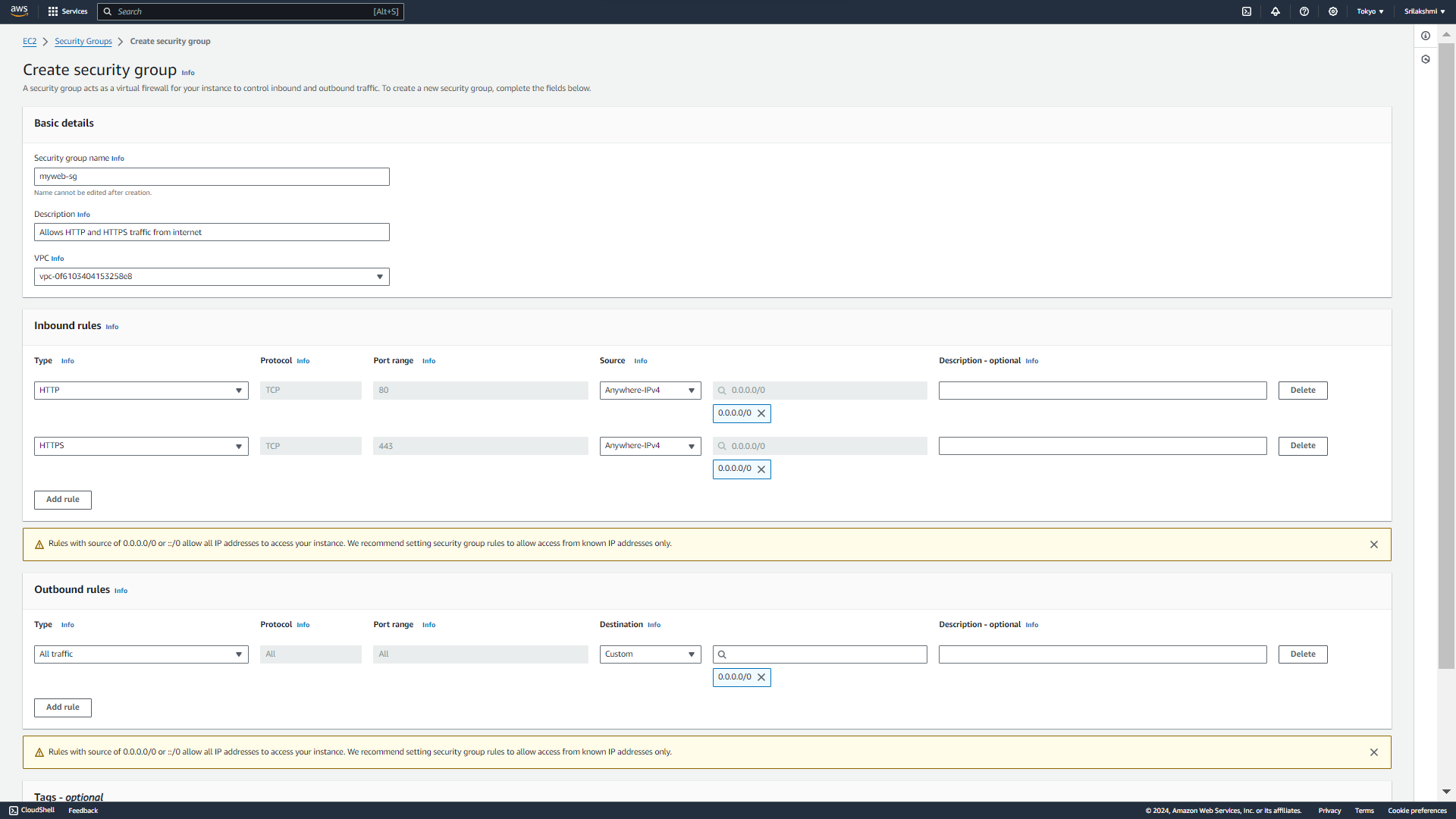


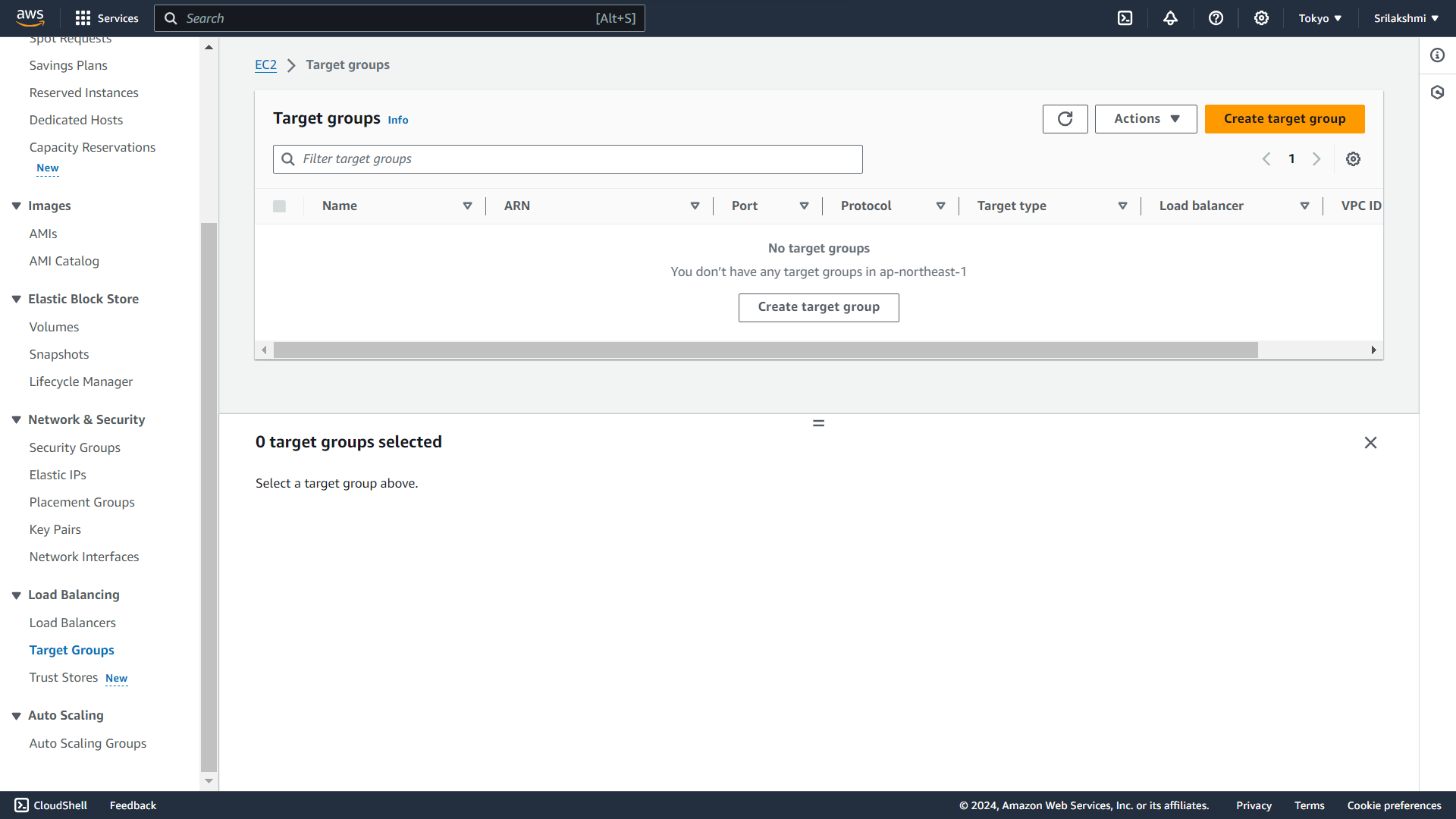
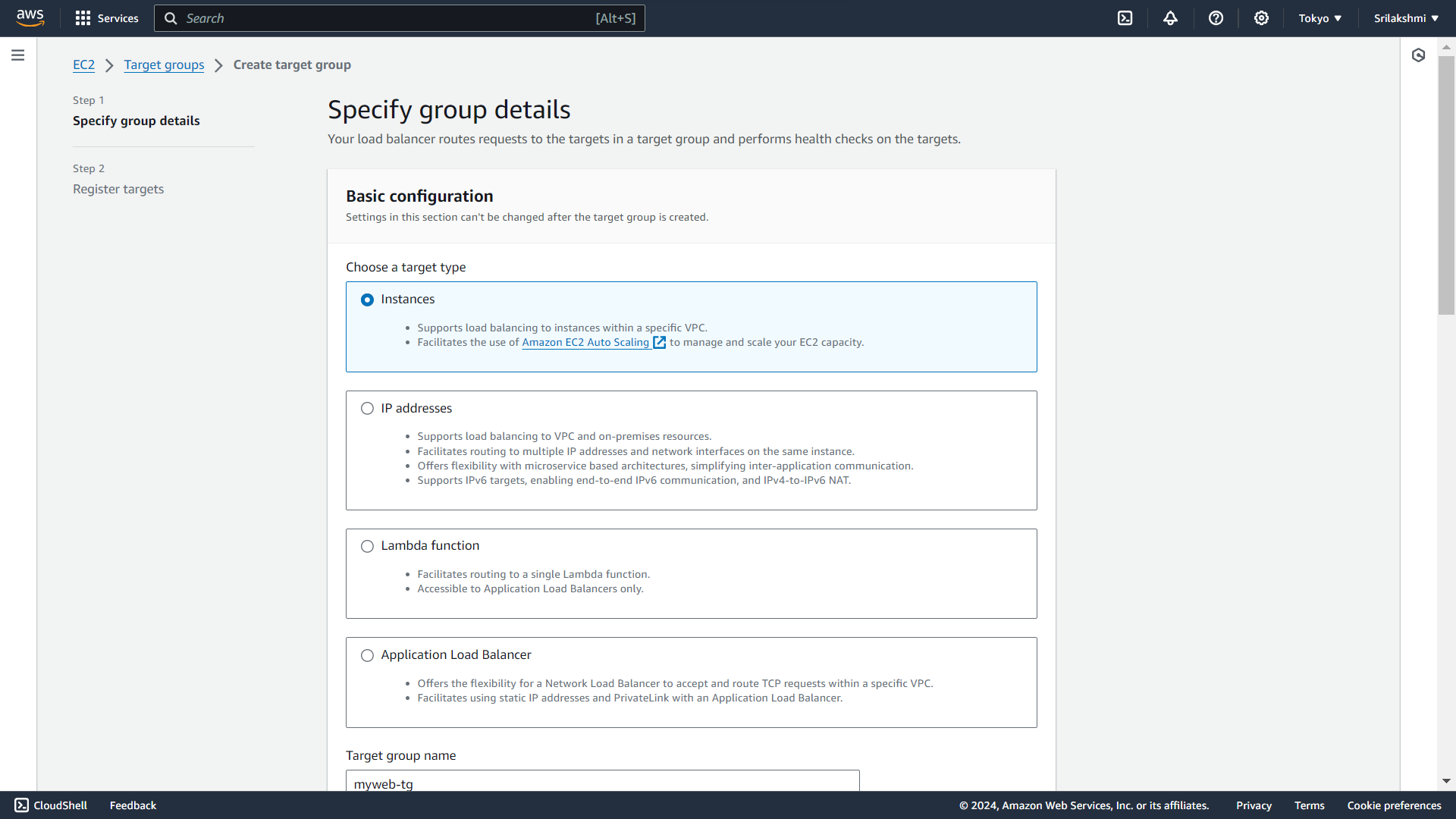
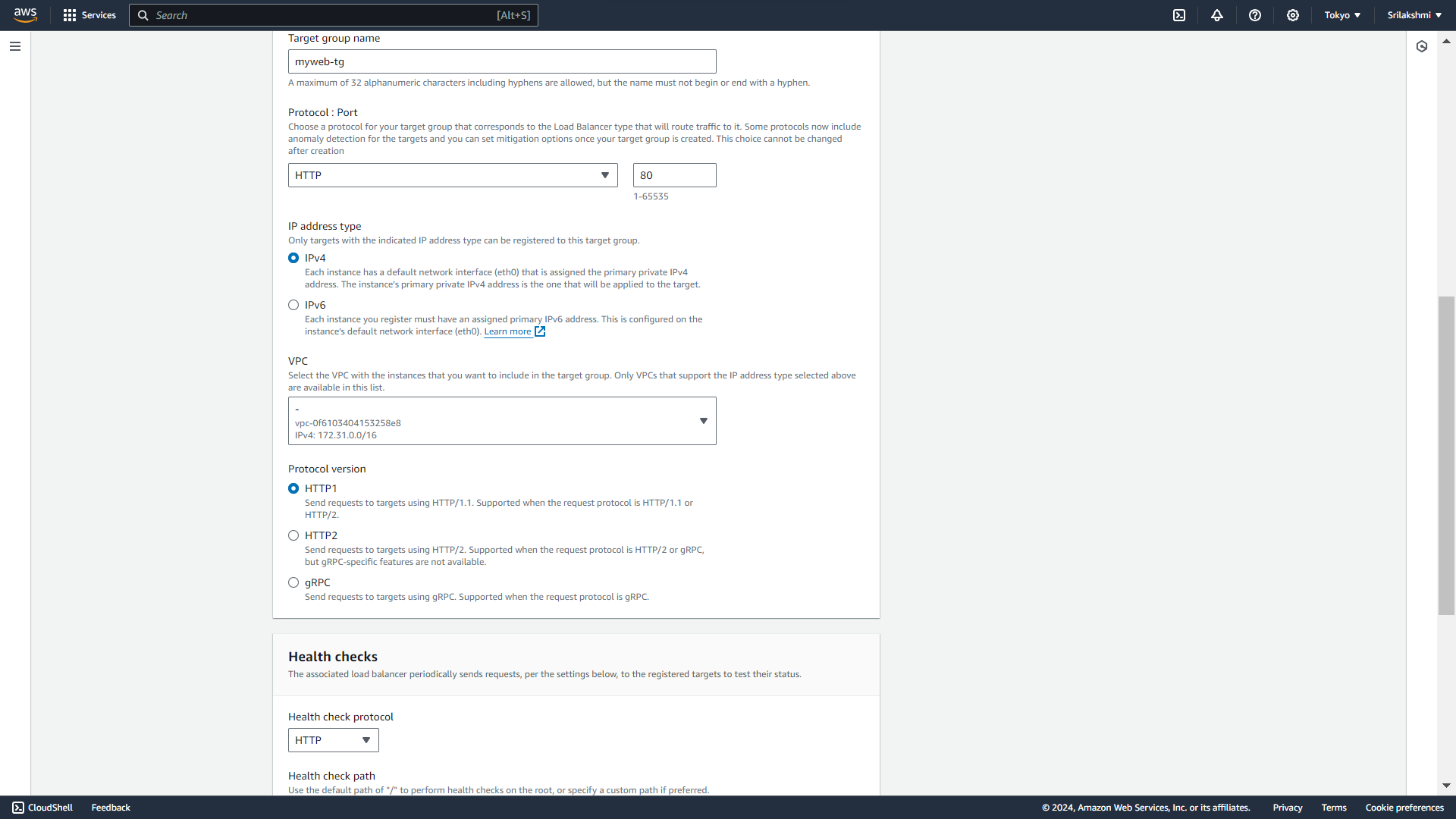
**Select 1st instance and copy the public IP address**

  
**Paste the copied public IP address on the browser it will access the application**

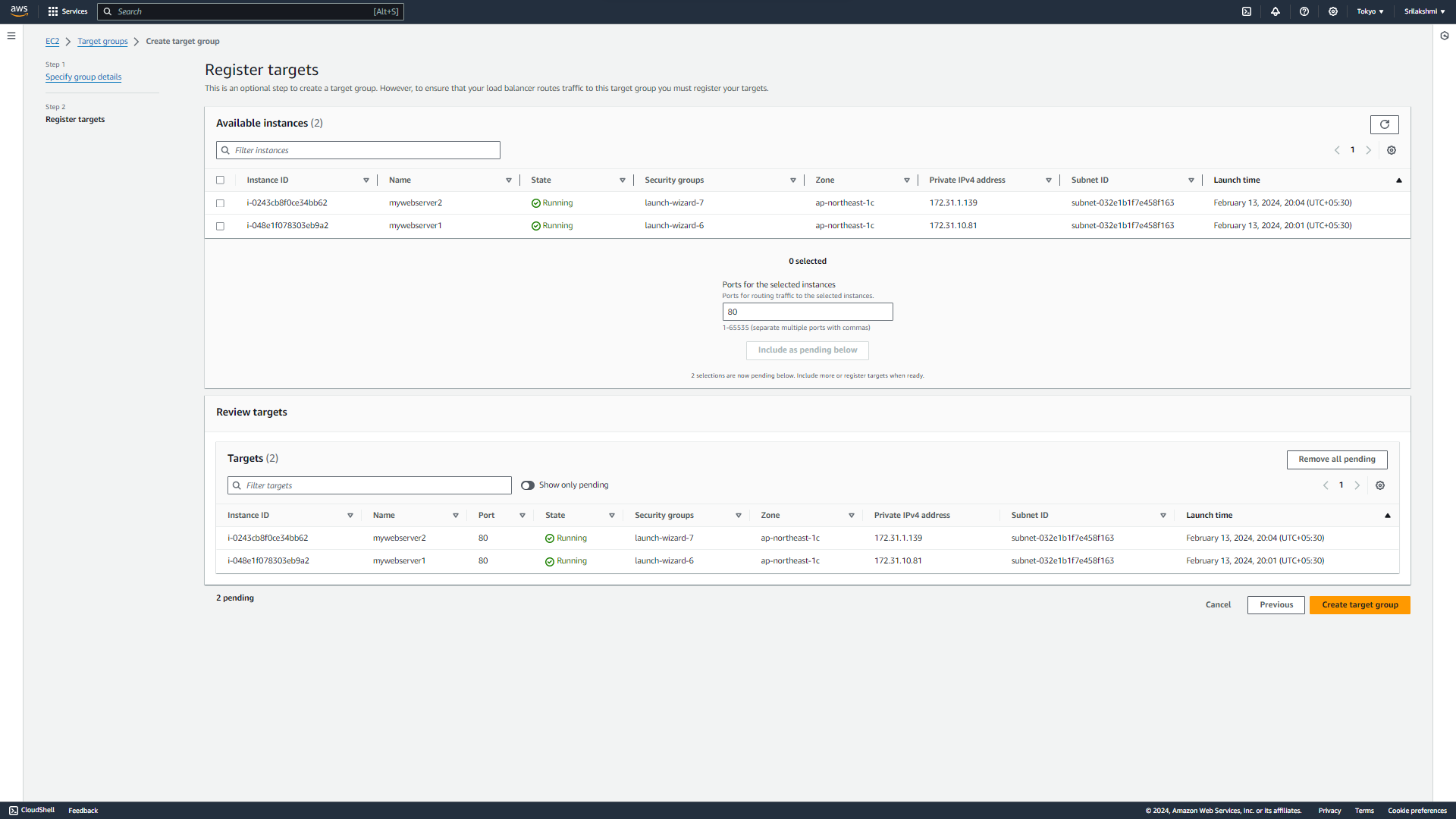


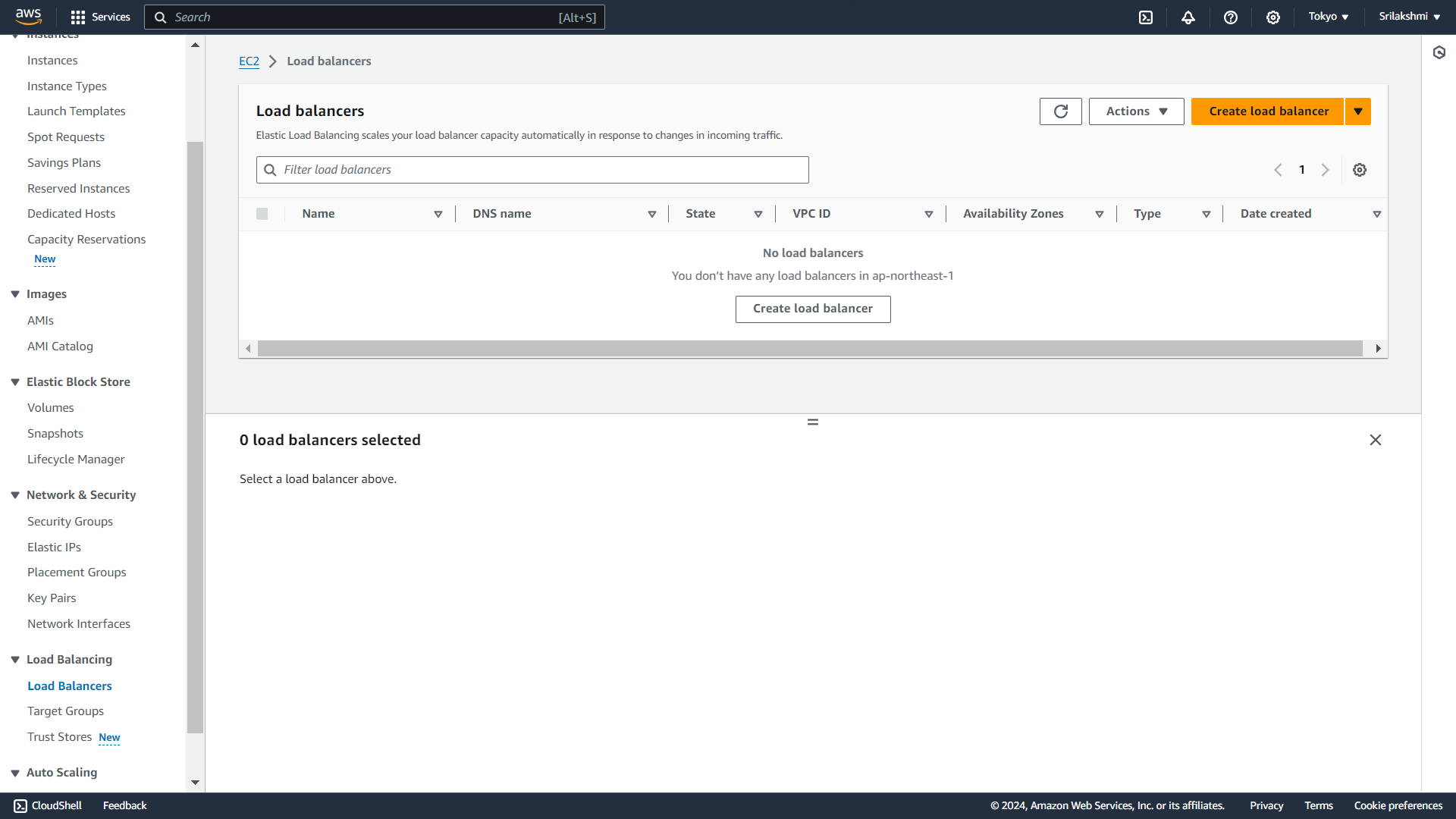
**Select 2nd instance and copy the public IP address**

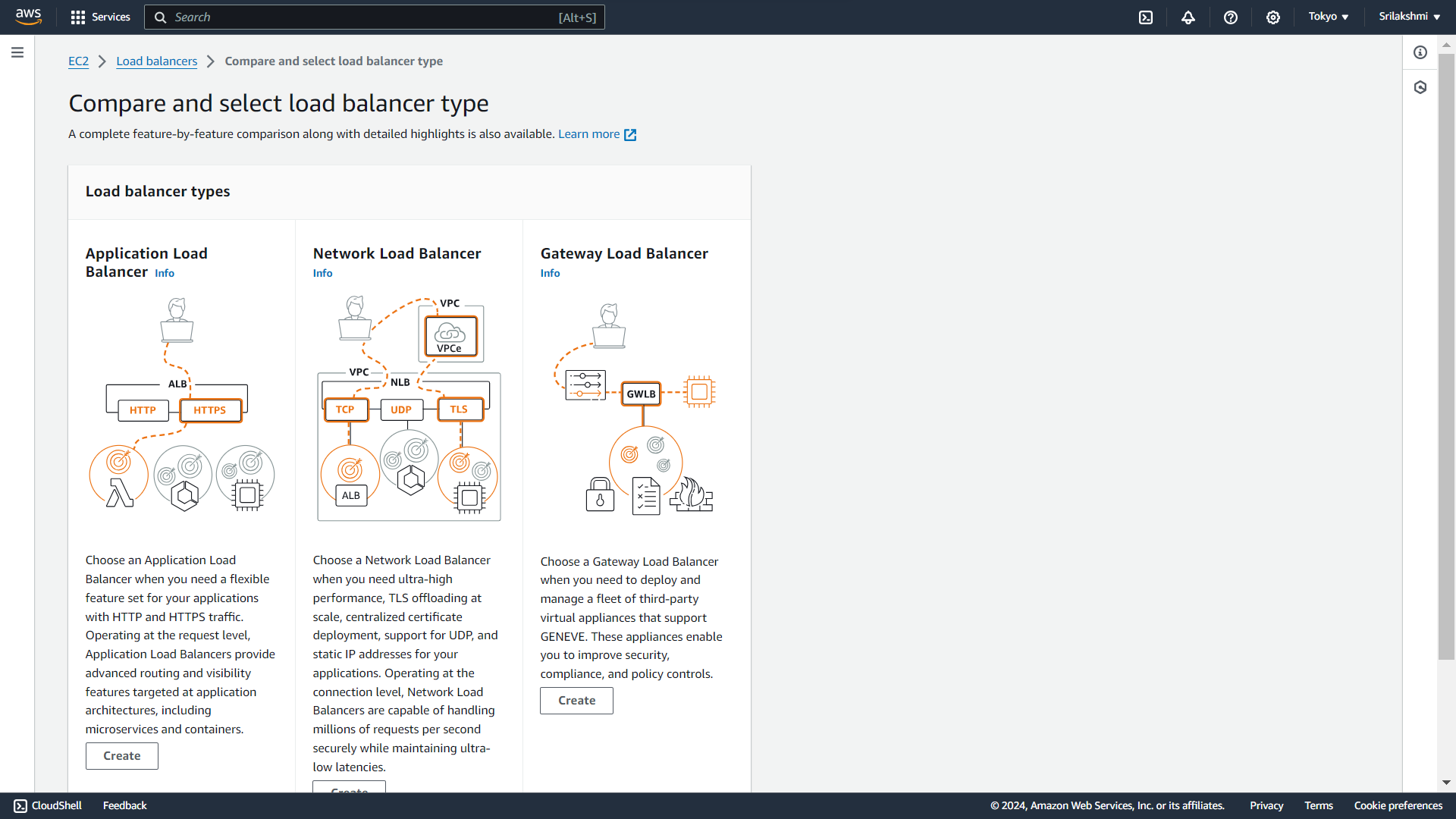
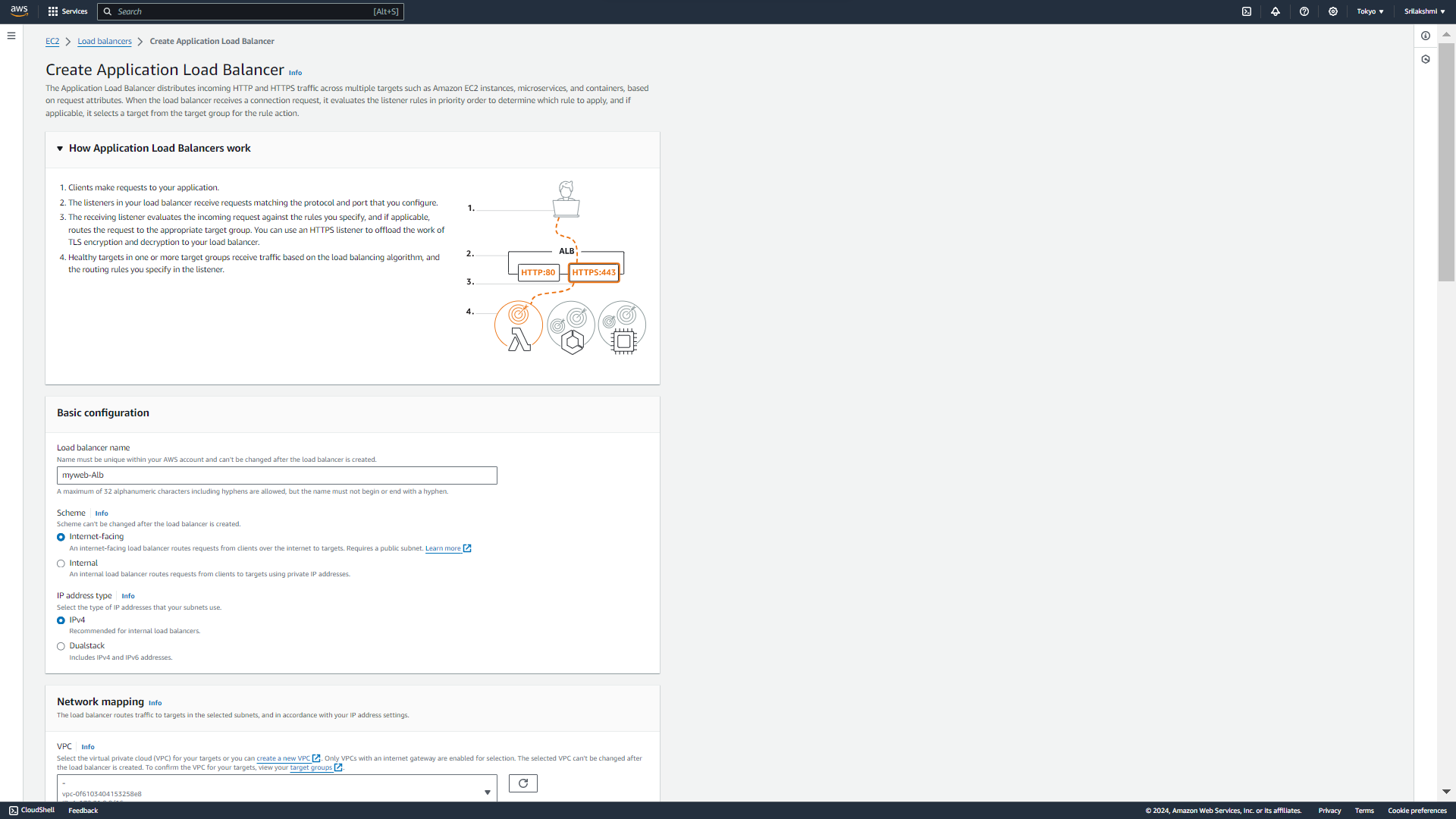
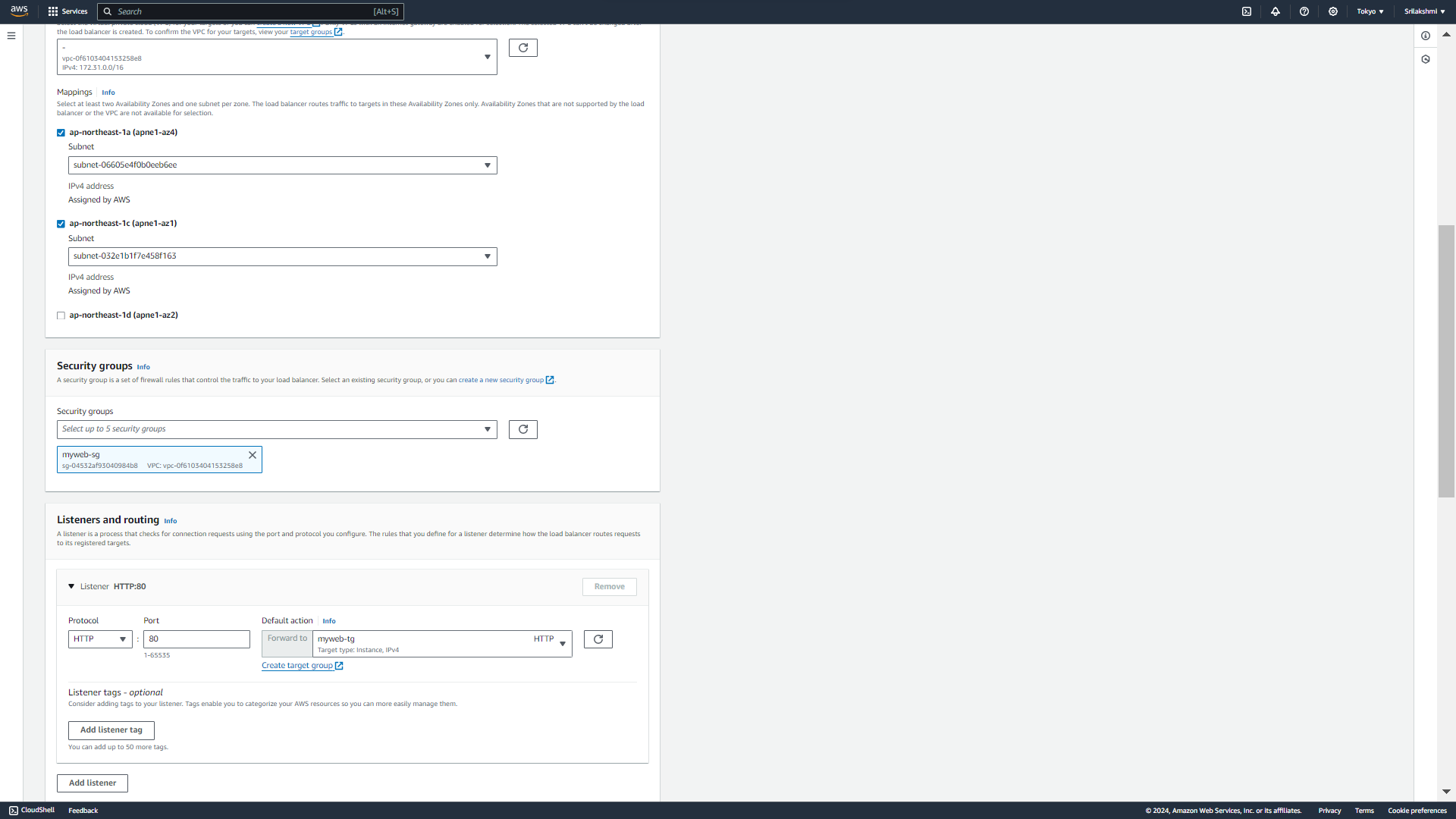
  
**Paste the copied public IP address on the browser it will access the application**  
 **Go to security groups and click on create security group**  
**Create security groups-myweb-sg  
edit inbound rules- Allow traffic from HTTP and HTTPS**  
  
**Go to target group and click on create target group**

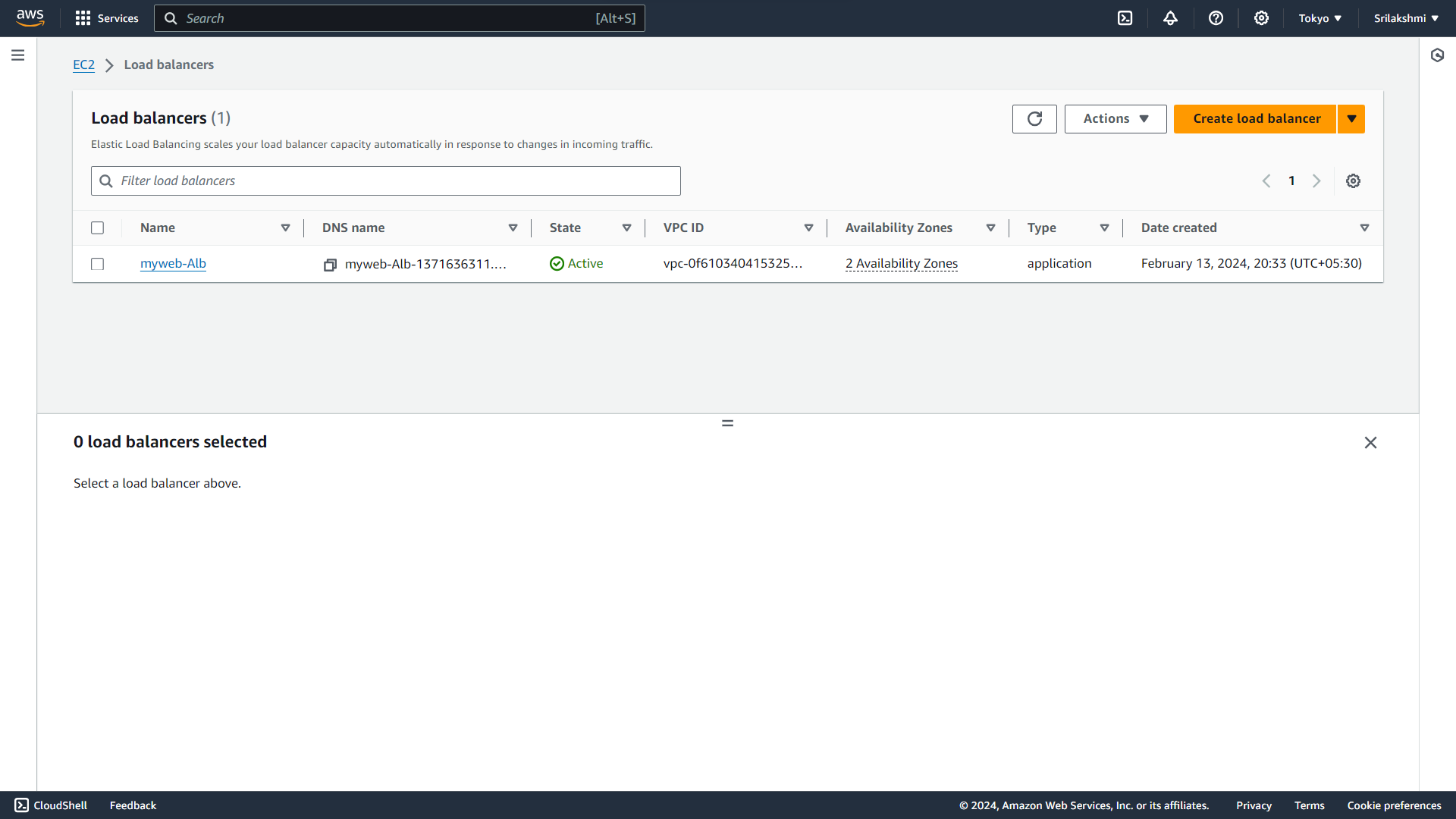
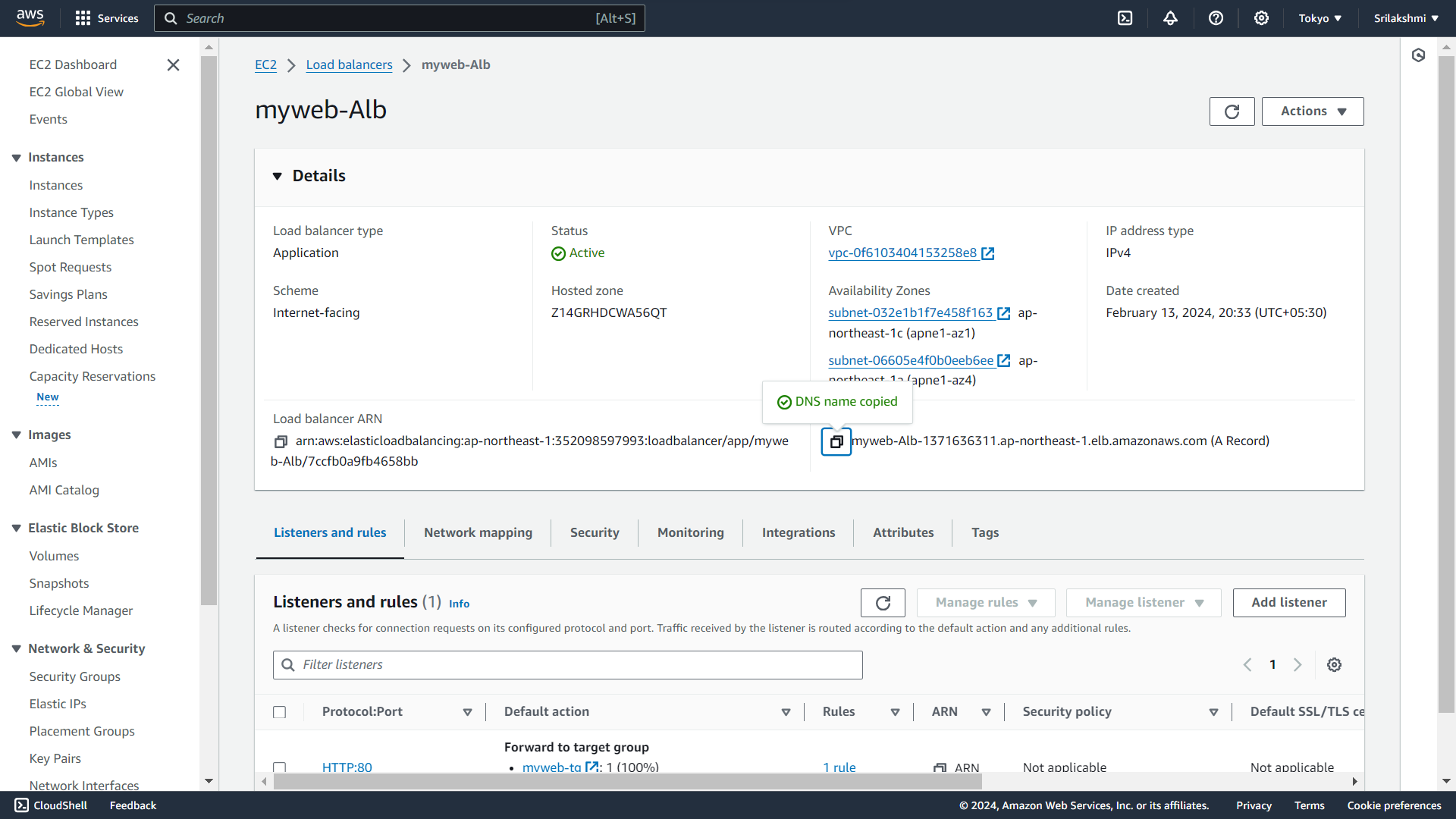
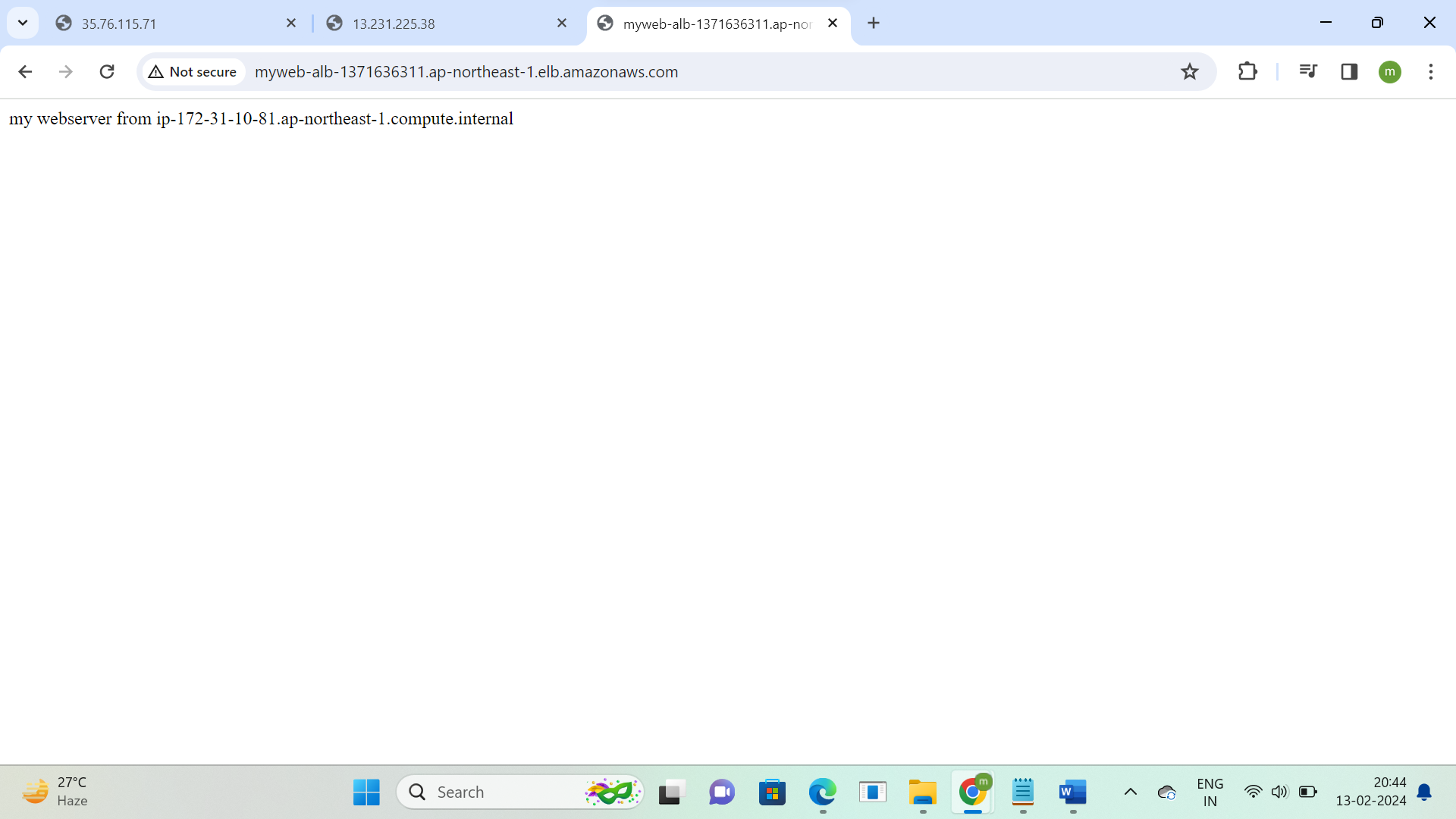
  
**Select instances type**  
  
**Give target groups-myweb-tg, select default vpc, protocol-HTTP, port-80  
IP address-IPV4, protocol version-HTTP1 and click on next**  


**Select target groups, click on include and pending below and click on create target group**

  
**Now go to Load balancer and click on create load balancer**

  
**Select Application load balancer**

  
**Give Load balancer name-myweb-Alb, select default vpc**  
**Select 2 availability zones, select created security group and target group   
Click on create load balancer**  
  
  
**This is the created Load balancer**

  
**Copy the DNS name of the Load Balancer**  
  
**Paste the copied DNS name in the browser 1st it will access the 1st instance**   
  
**When we refresh it will access the 2nd instance**

