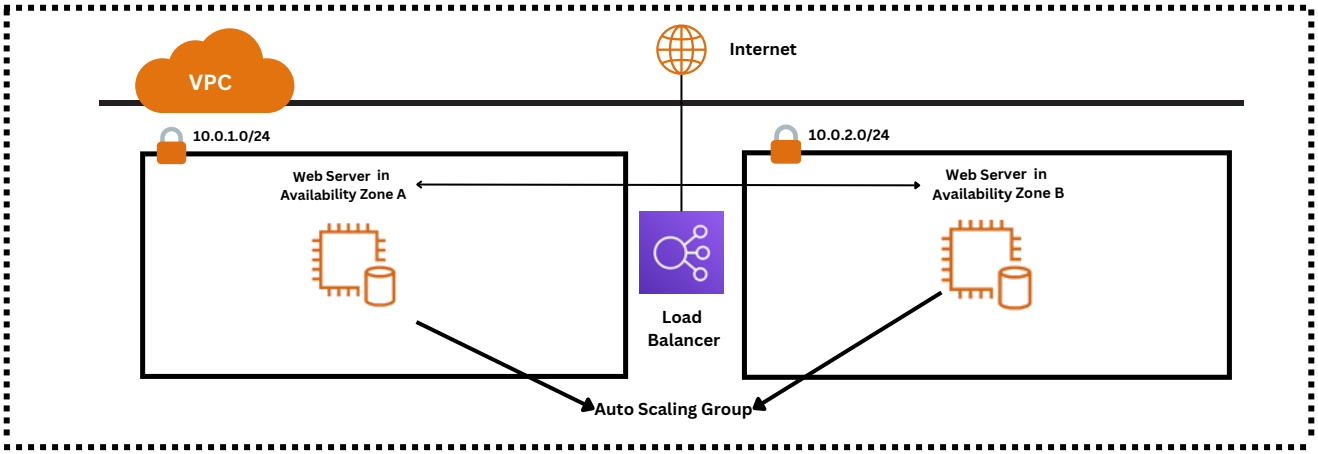
Elastic Load Balancing & Auto Scaling Groups Section



Scalability & High Availability

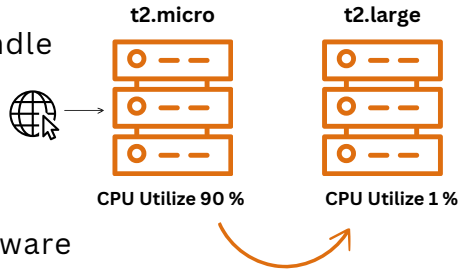
Scalability means a system can handle more work when needed.

- **Vertical Scalability** – Upgrading a single server with more RAM and a faster processor (like upgrading your laptop to a more powerful one)
- **Horizontal Scalability** –Adding more servers to distribute the workload (like using multiple laptops to process tasks together).

Vertical Scalability

Vertical Scalability means making a single server more powerful by upgrading its hardware.

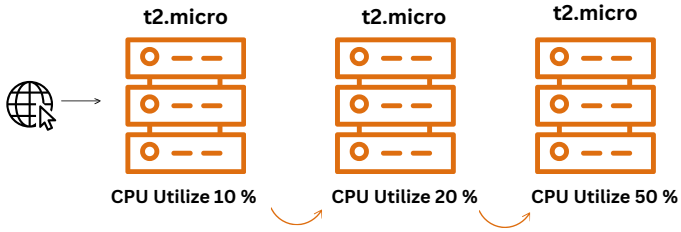
- **Example:** Your application runs on a **t2.micro server**. To handle more load, you upgrade it to a **t2.large server**.
- This type of scaling is common for systems that aren't distributed, like databases.
- **Limitation:** There's a maximum upgrade limit based on hardware capacity.



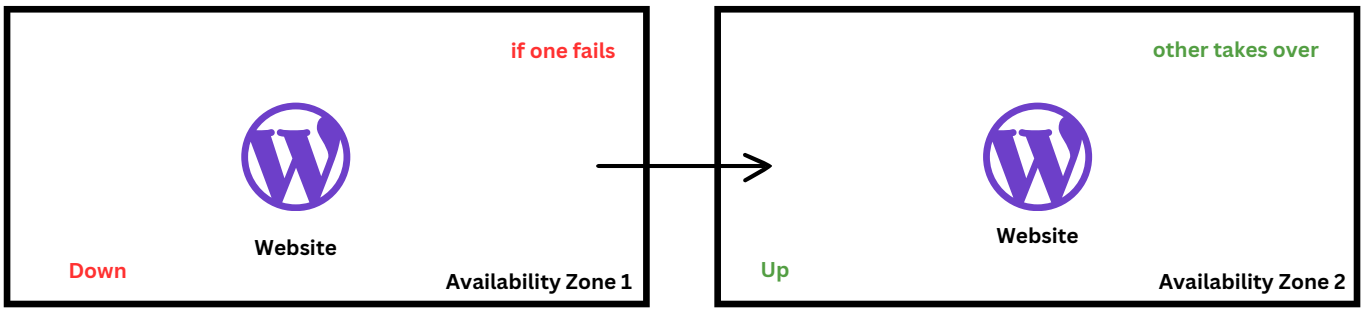
Horizontal Scalability

Horizontal Scalability means adding more servers to handle more traffic or workload

- **Example:** Instead of upgrading a single server with more RAM, you add multiple servers to share the work.
- This method is common for web applications and modern systems that need to handle many users.
- Cloud services like Amazon EC2 make it easy to add more servers when needed.



High Availability



Running your application on two or more servers in different locations

High Availability

High Availability means keeping your system running even if one part fails.

- It usually works with horizontal scaling (multiple servers).
- **Example:** Running your application on two or more servers in different locations, so if one fails, the other takes over.
- The goal is to prevent downtime, even if a data center has a problem or stops working.

High Availability & Scalability for EC2 (Simple Explanation)

Vertical Scaling: Make a single server bigger (scale up or down).

- **Example:** Start with a small server (**like t2.nano with 0.5GB of RAM**) and upgrade to a much larger server as your needs grow.

Horizontal Scaling: Add more servers to handle more traffic (scale out or in).

- **Example:** Use Auto Scaling Groups to automatically add or remove servers.
- Use a Load Balancer to distribute the work evenly across all servers.

High Availability: Run your servers in multiple locations (Availability Zones) to ensure the system keeps running, even if one location fails.

- **Example:** Set up your **Auto Scaling Group** and **Load Balancer** to work across **multiple Availability Zones** for better reliability and uptime.

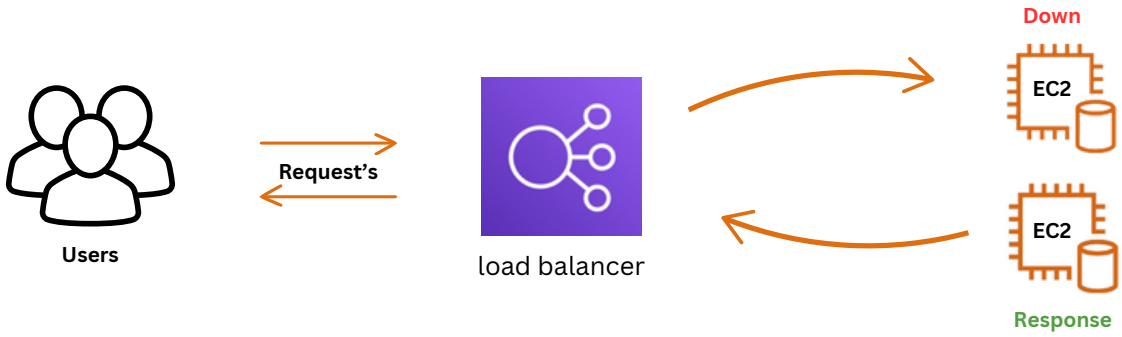
Scalability vs Elasticity vs Agility

- **Scalability:** The ability to handle more work by either upgrading your existing system (making it faster or stronger) or adding more devices (like getting more cash registers in a store).
- **Elasticity:** Once your system is scalable, elasticity means it can automatically change based on demand. For example, if there are more customers (like during sales), the system can add more cash registers, and when it's quieter, it can remove them. This helps save costs because you only pay for what you need
- **Agility:** Agility is the ability to quickly get new resources when needed. For example, in the cloud, you can add new computers for your team within minutes instead of waiting days or weeks. (This is more about speed, not directly about scaling).

load balancing

Load balancers are like traffic managers that direct website visitors to different servers (computers) so no single server gets too busy..

- **For example,** imagine a popular restaurant with several chefs in the kitchen. The load balancer is like the host who guides customers to different chefs based on who is least busy, ensuring that no chef gets overwhelmed



- **Distribute traffic:** It sends website visitors to different servers so no server gets too crowded.
- **Single address:** It gives users one easy link (like a website URL) to reach your app, no matter which server is actually handling it.
- **Handle server issues:** If one server stops working, the load balancer sends traffic to other working servers.
- **Health checks:** It regularly checks if the servers are working and stops sending traffic to broken ones.
- **Secure websites:** It keeps websites safe by managing secure connections (HTTPS).
- **Stay available:** The system keeps working even if one server or location has problems by using backup servers in different places.

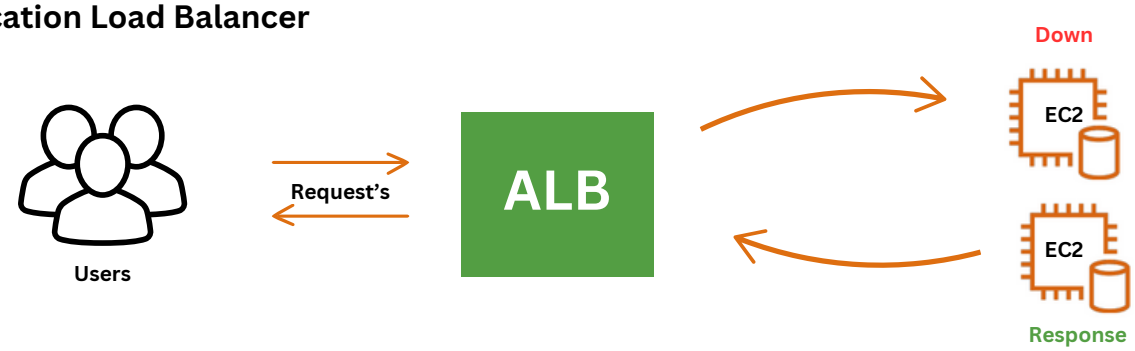
Elastic Load Balancer (ELB)

- **Managed service:** An ELB is a service that AWS manages for you, so you don't need to worry about running it yourself
- **Reliable:** AWS ensures the ELB is always working and takes care of updates and maintenance.
- **High availability:** AWS makes sure the ELB is always available and ready to use.
- **Easy setup:** AWS provides a simple setup with only a few settings to adjust.

Types of ELBs AWS

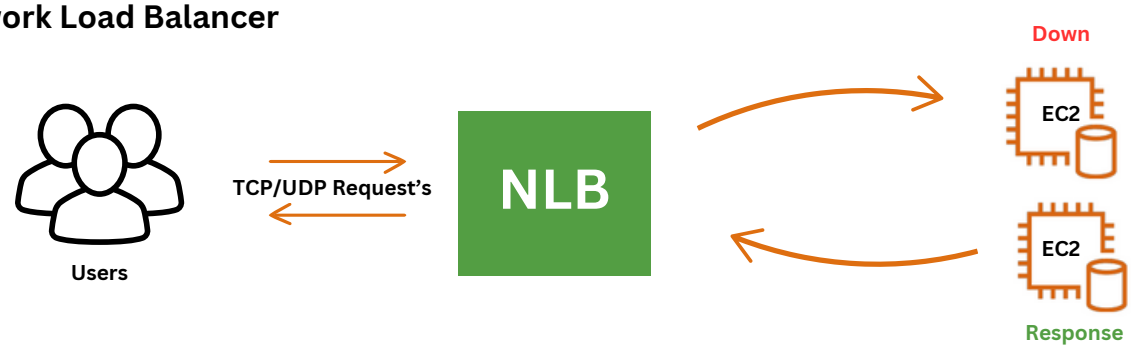
- **Application Load Balancer (Layer 7):** Works with HTTP/HTTPS traffic.
- **Network Load Balancer (Layer 4):** High-performance option that works with TCP traffic.
- **Gateway Load Balancer (Layer 3):** For special types of traffic routing.
- **Classic Load Balancer (Layer 4 & 7):** Retired in 2023, used to handle both types of traffic.

Application Load Balancer



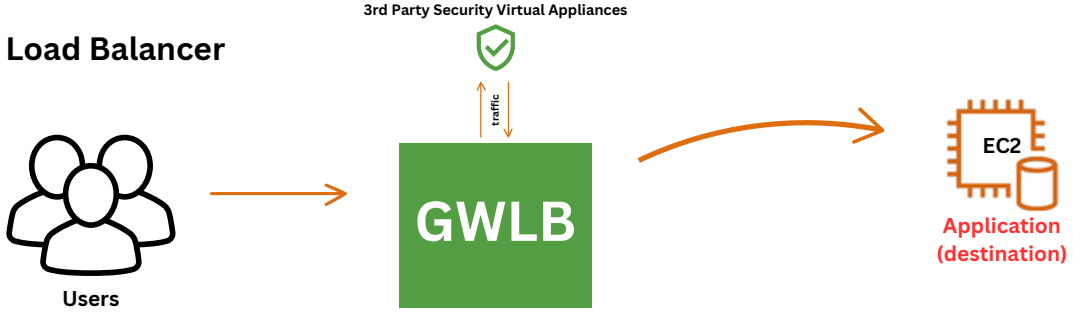
- **Supports HTTP, HTTPS, and gRPC:** The load balancer works with common web protocols like HTTP, HTTPS (for secure websites), and gRPC (used for high-performance communication between apps).
- **HTTP Routing:** It can direct traffic based on specific rules, such as URL paths or domain names, to the correct server.
- **Static DNS:** It provides a fixed web address (URL) that doesn't change, so users can always reach your application

Network Load Balancer



- **Supports TCP and UDP:** The load balancer works with low-level network protocols like TCP and UDP, which are commonly used for real-time applications like gaming or video streaming.
- **High Performance:** It can handle millions of requests per second, making it suitable for very fast, high-traffic applications.
- **Static IP:** It provides a fixed IP address through Elastic IP, so your app always has the same address, even if the underlying infrastructure changes.

Gateway Load Balancer

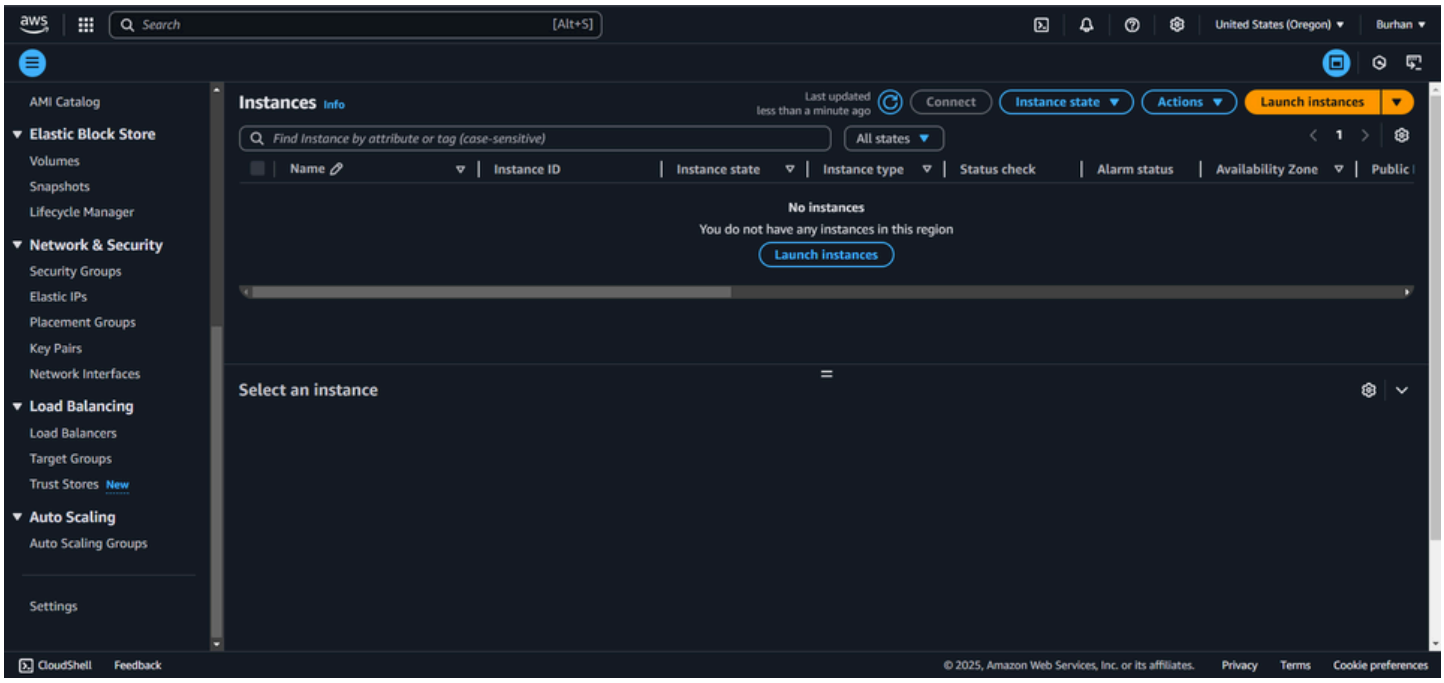


- **Routes traffic to firewalls:** It can send internet traffic to security systems (firewalls) that you control on EC2 instances to protect your app.
- **Intrusion detection:** It helps detect any unwanted or harmful activities trying to access your system, keeping your app secure.

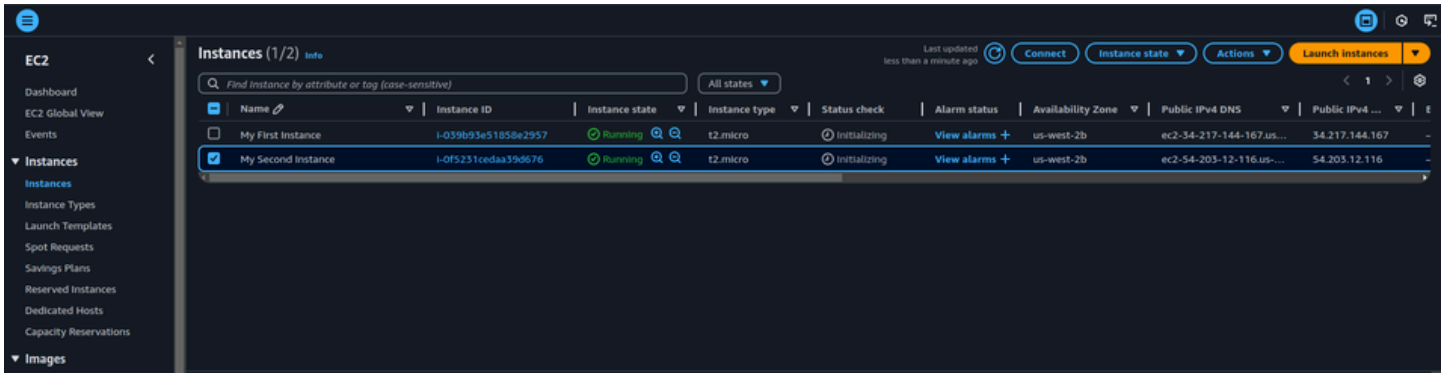
Application Load Balancer ALB Hands On

Step 1 : Delete All Previous Configuration

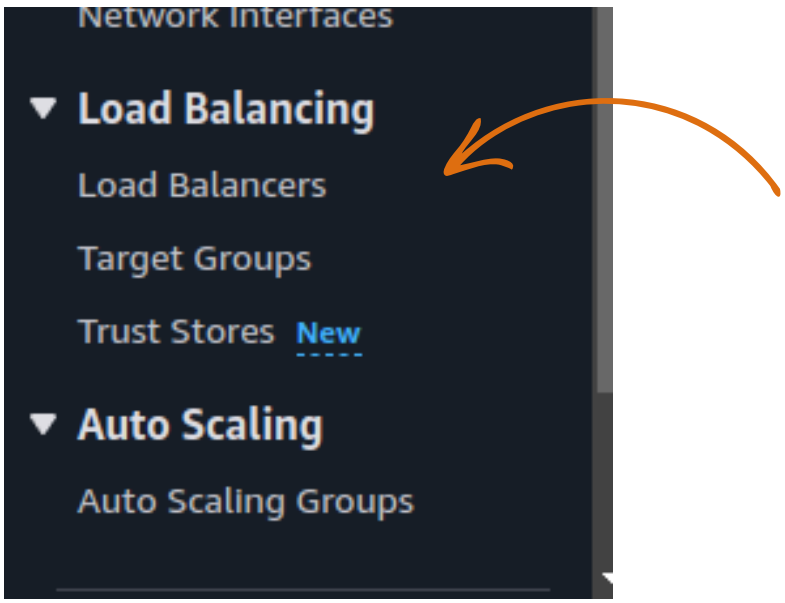
- Delete Instance
- Delete Group Policy



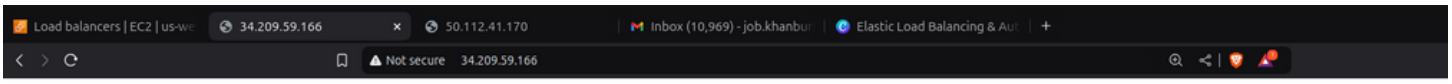
Create Two Instance



Create Load Balancer

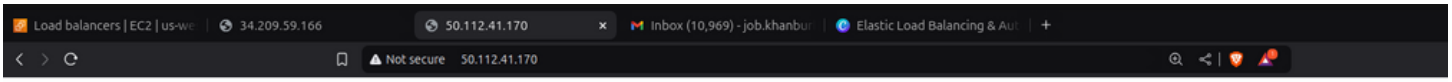


Access Web Page Through Instance Live IP Address



Hello Friend, This is Server One ip-172-31-23-78.us-west-2.compute.internal

Server Second

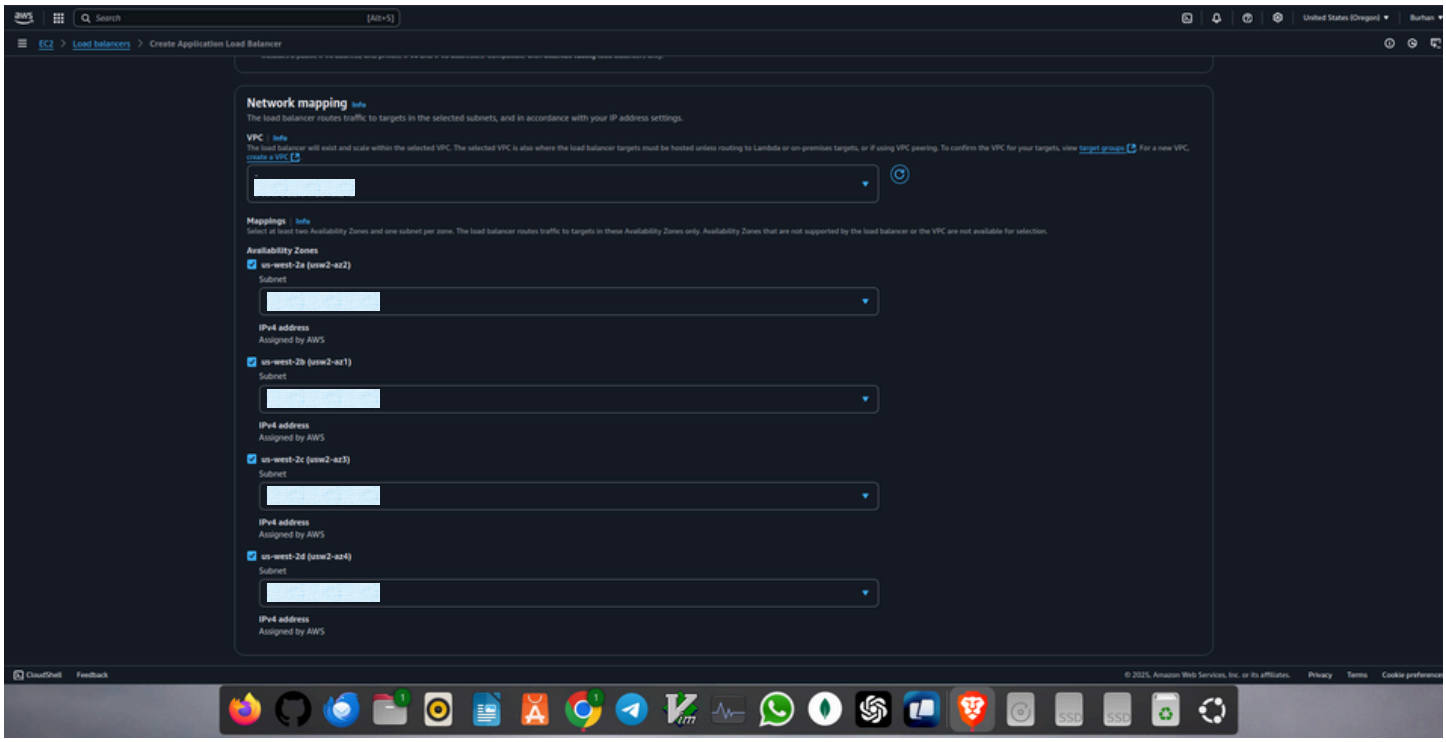


Hello Friend, This is Server One ip-172-31-28-110.us-west-2.compute.internal

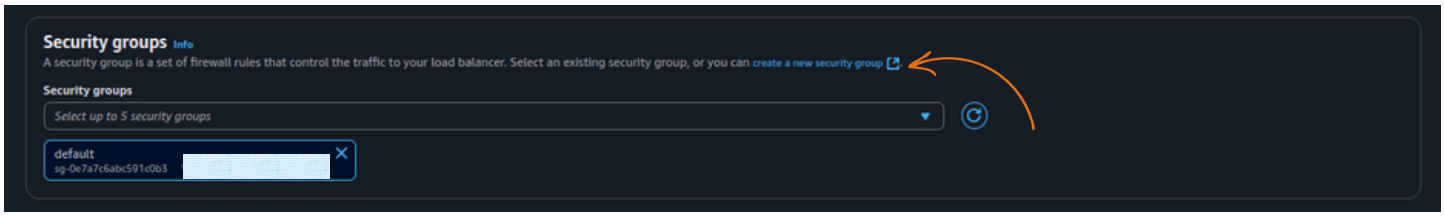
Create Load Balancer

Create Load Balancer

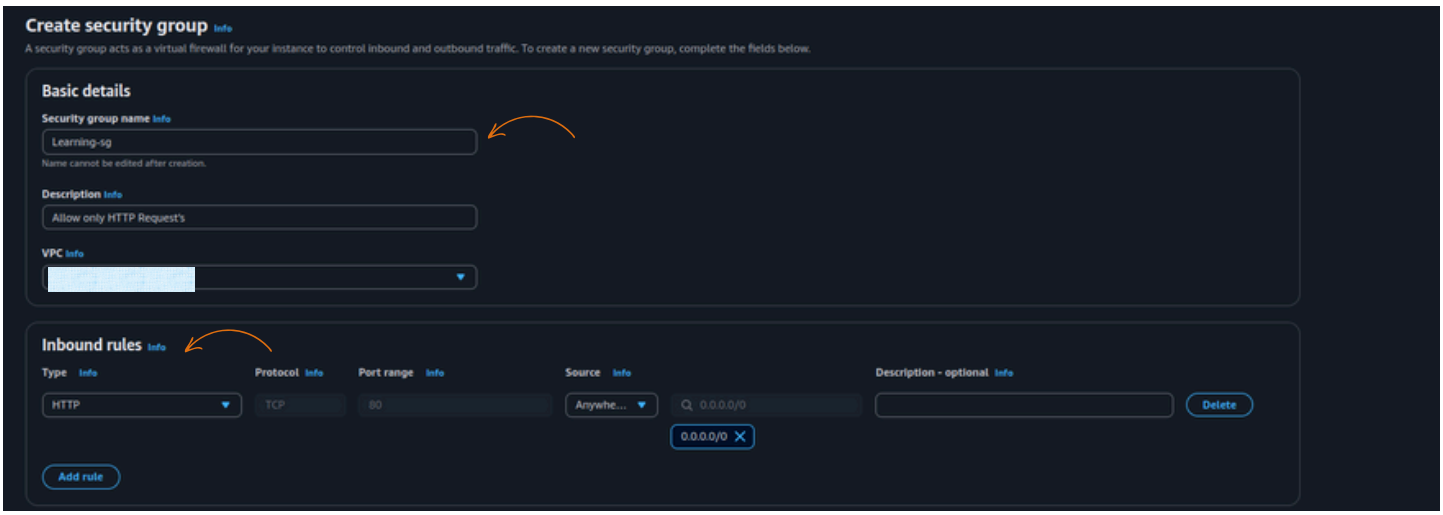
Network Mapping Tick All Availability Zones

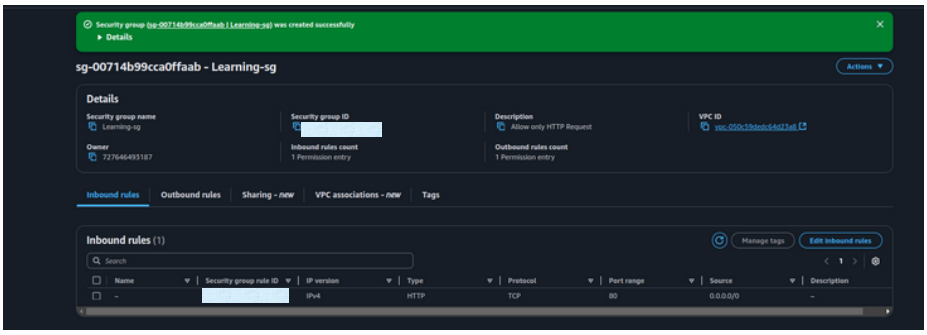


Create a new security group



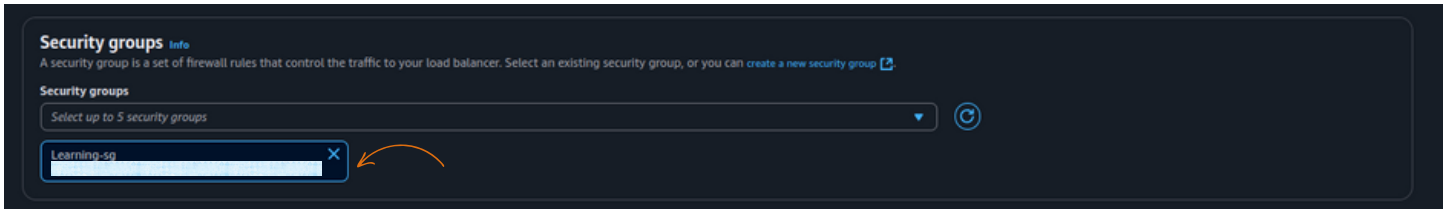
Create Security Group



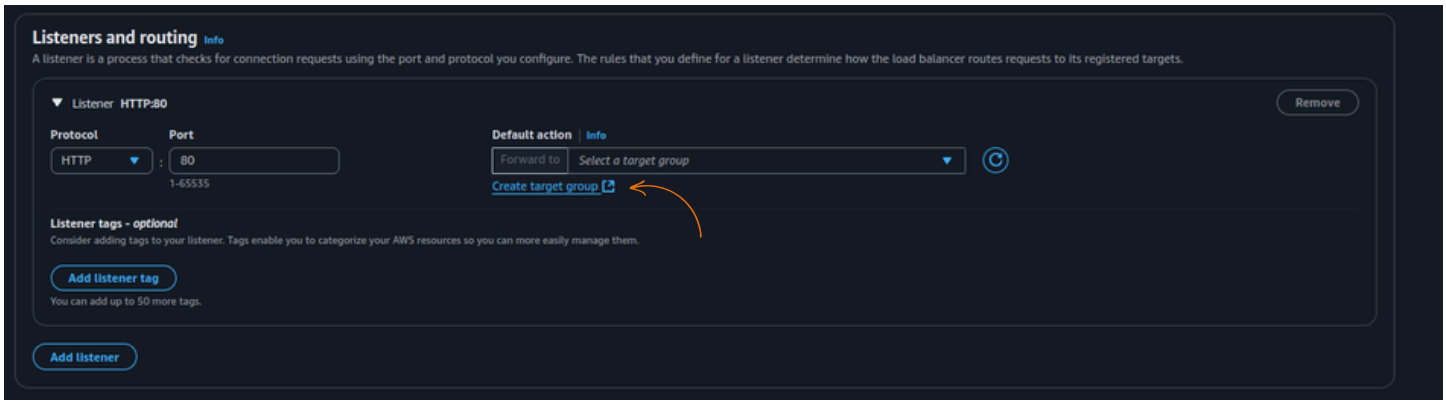


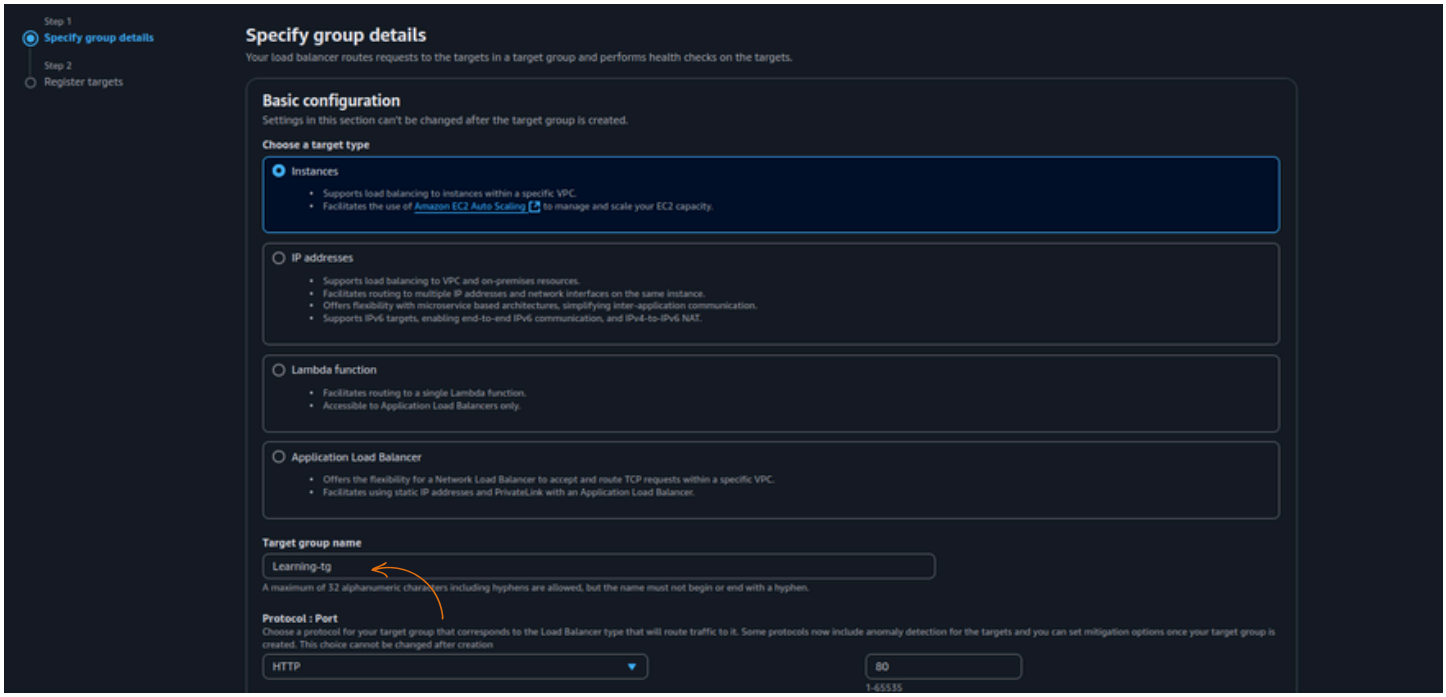
Create security group

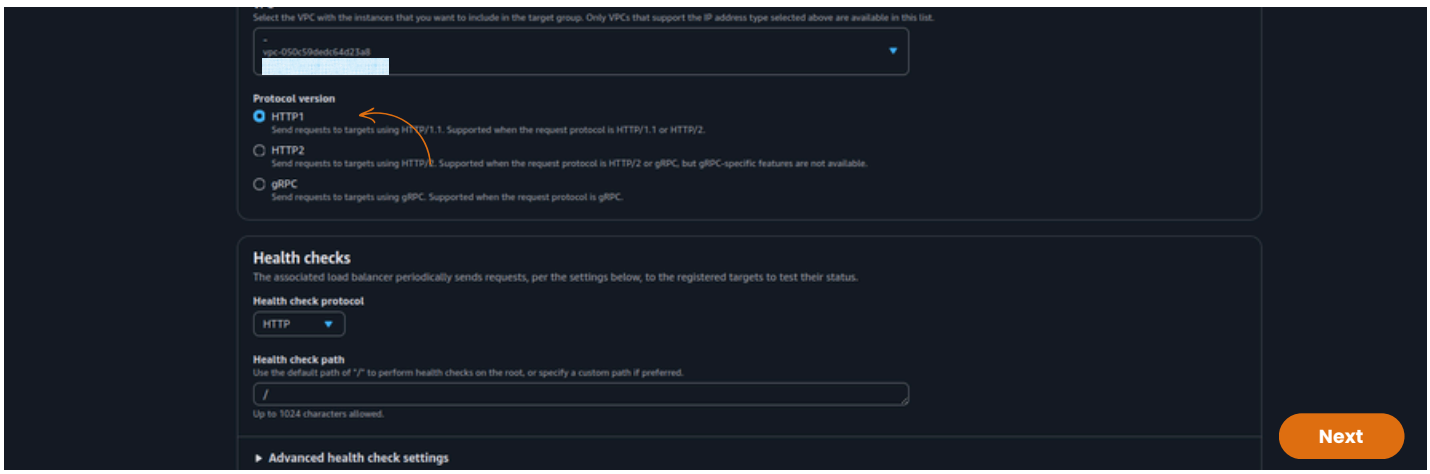
Select your Security Group : Learning-sg



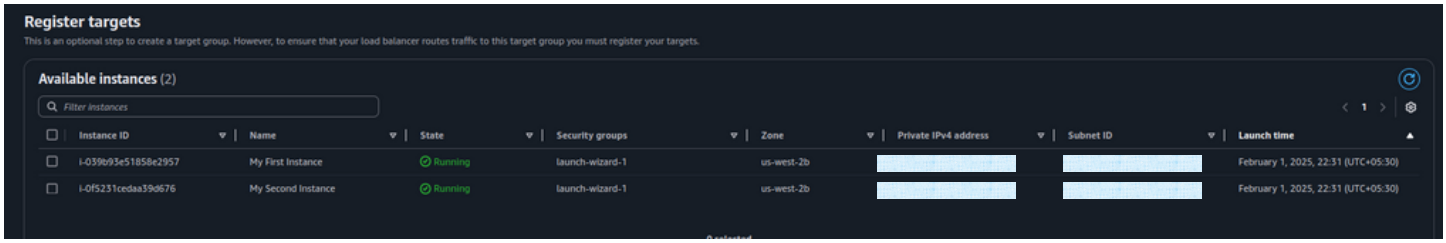
Listeners and routing : Create Target Group



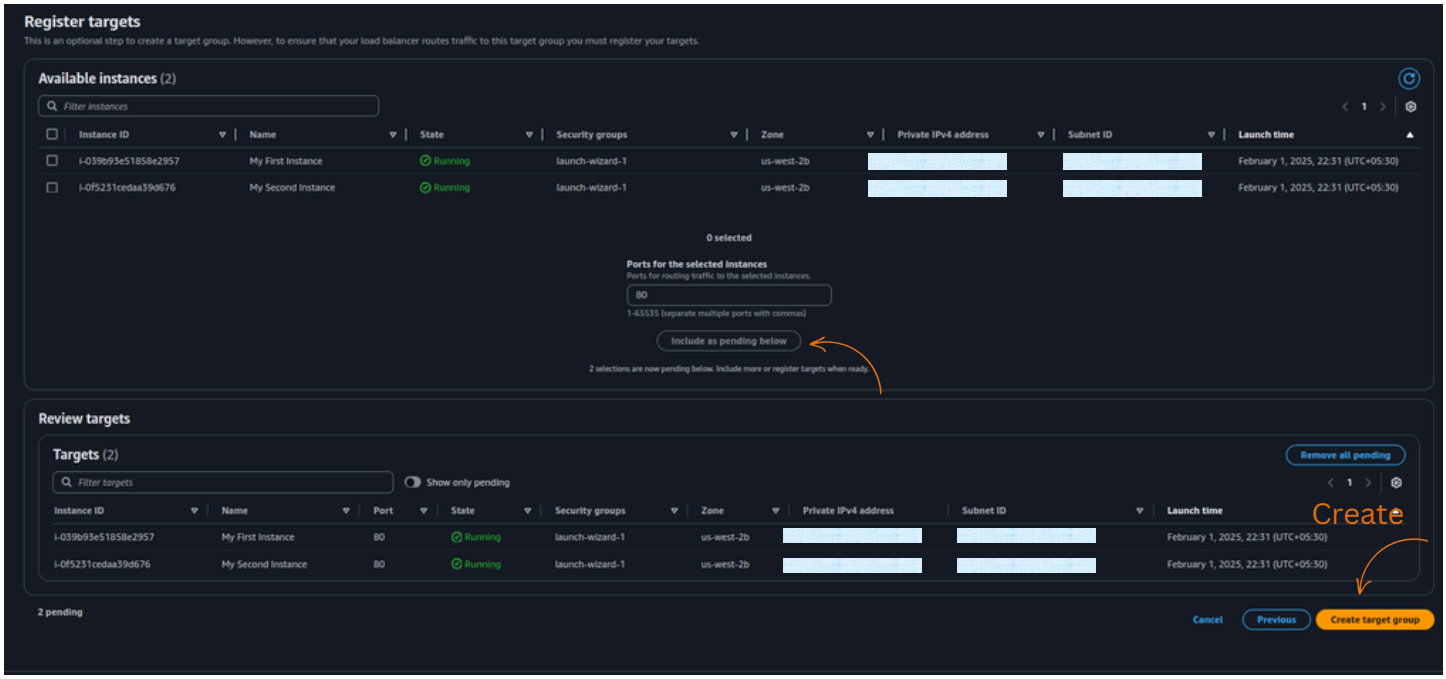




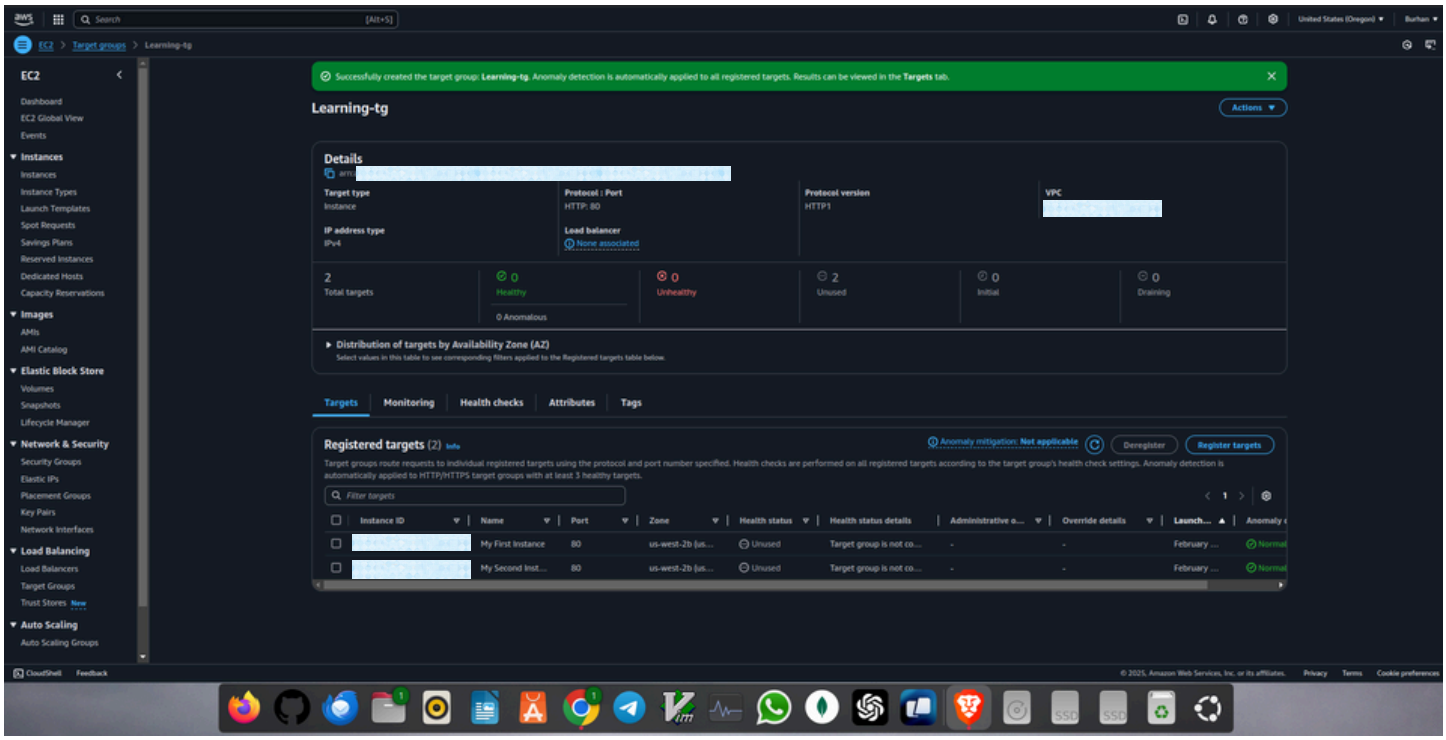
Register Targets



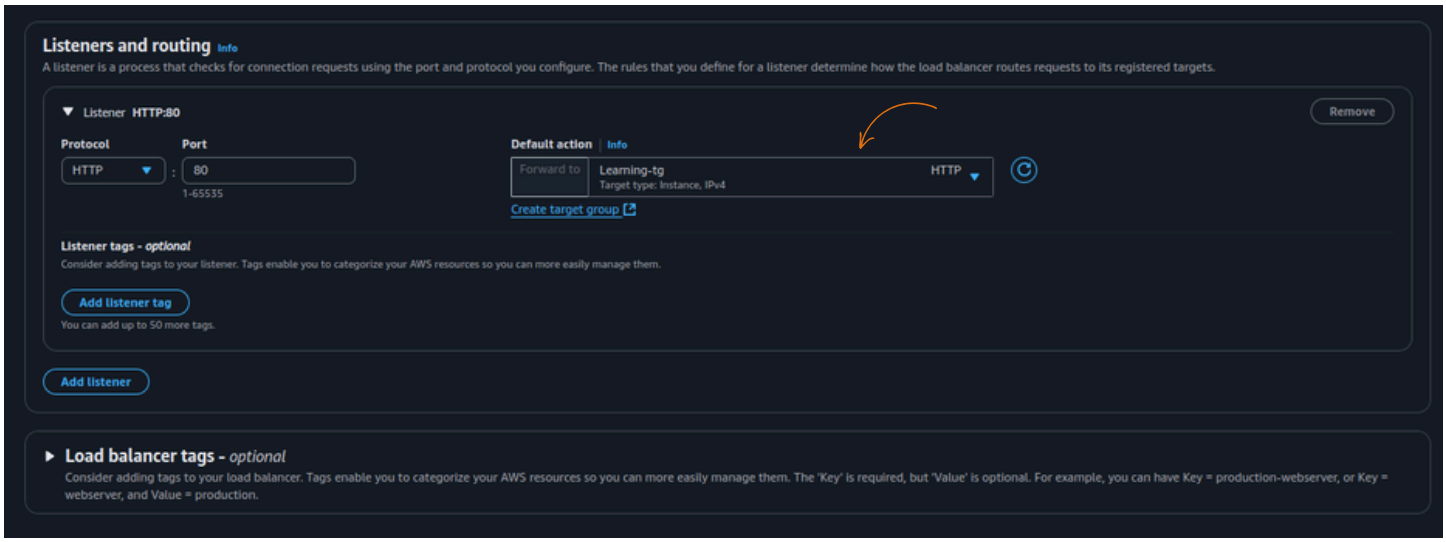
Check Both Instance id > click on Include as pending below

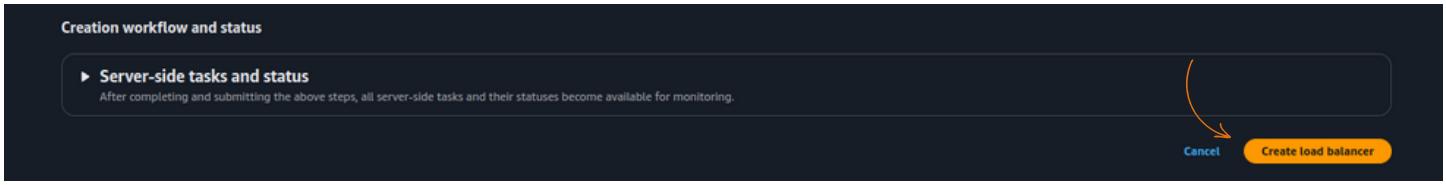


Target group: Learning-tg. Created



Add Target Group Here



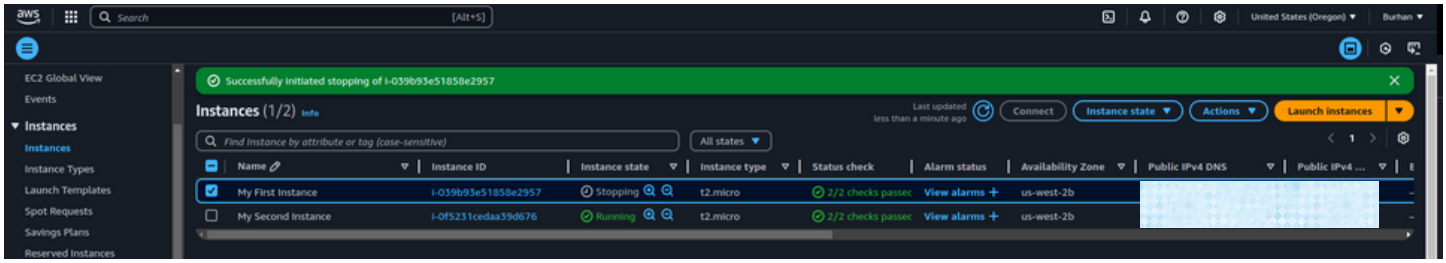


Copy DNS & Paste on Browser

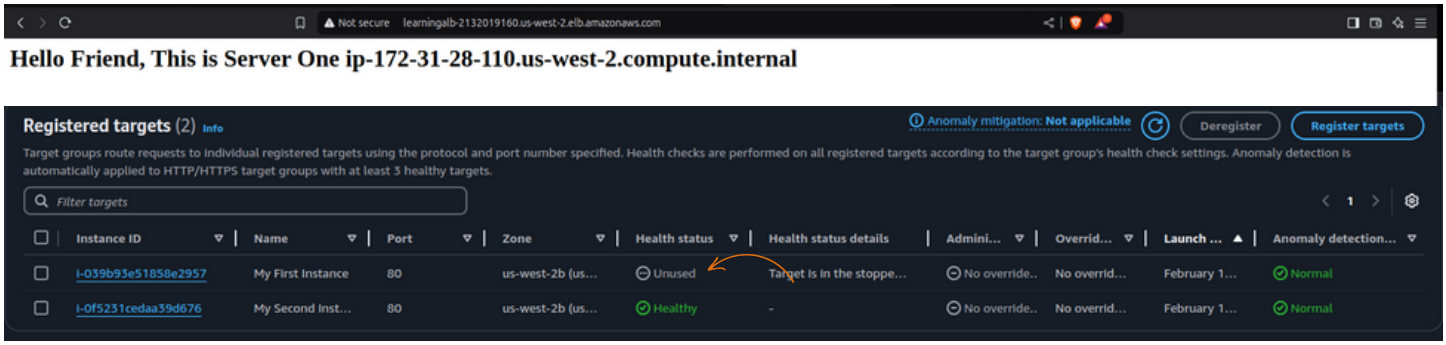
Response 1

Response 2

Stop One Instance



Till Website Running on Secondary Zone



Auto Scaling Group (ASG)

An Auto Scaling Group (ASG) is a feature in AWS that helps manage the number of EC2 instances (virtual servers) running for your application.

- Sometimes, your website or app gets a lot of traffic, and sometimes it doesn't.
- In AWS, you can easily add or remove servers based on the load.

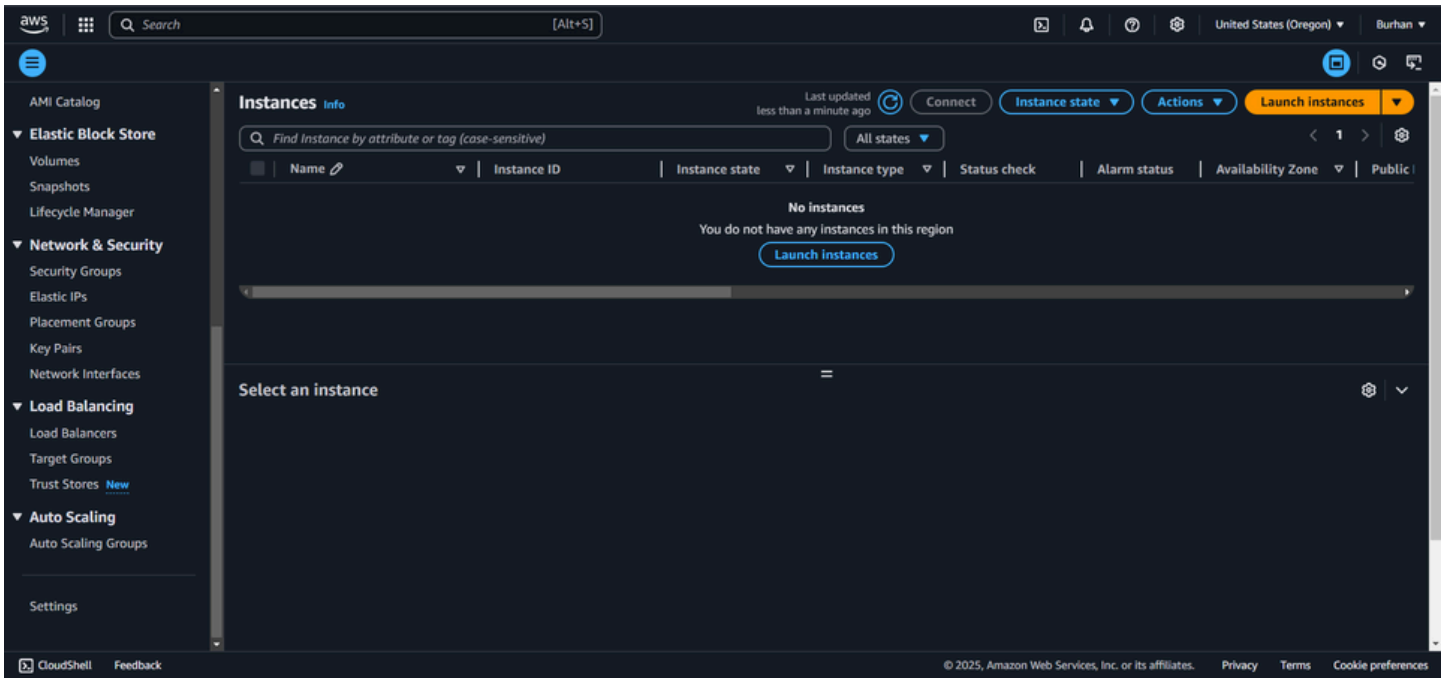
What does an Auto Scaling Group do ?

- Adds more servers (Scale Out) when traffic increases.
- Removes extra servers (Scale In) when traffic decreases.
- Keeps a minimum and maximum number of servers running at all times.
- Automatically connects new servers to a Load Balancer (which distributes traffic).
- Replaces unhealthy servers to keep the application running smoothly.
- Saves costs by ensuring you only use the number of servers you actually need.

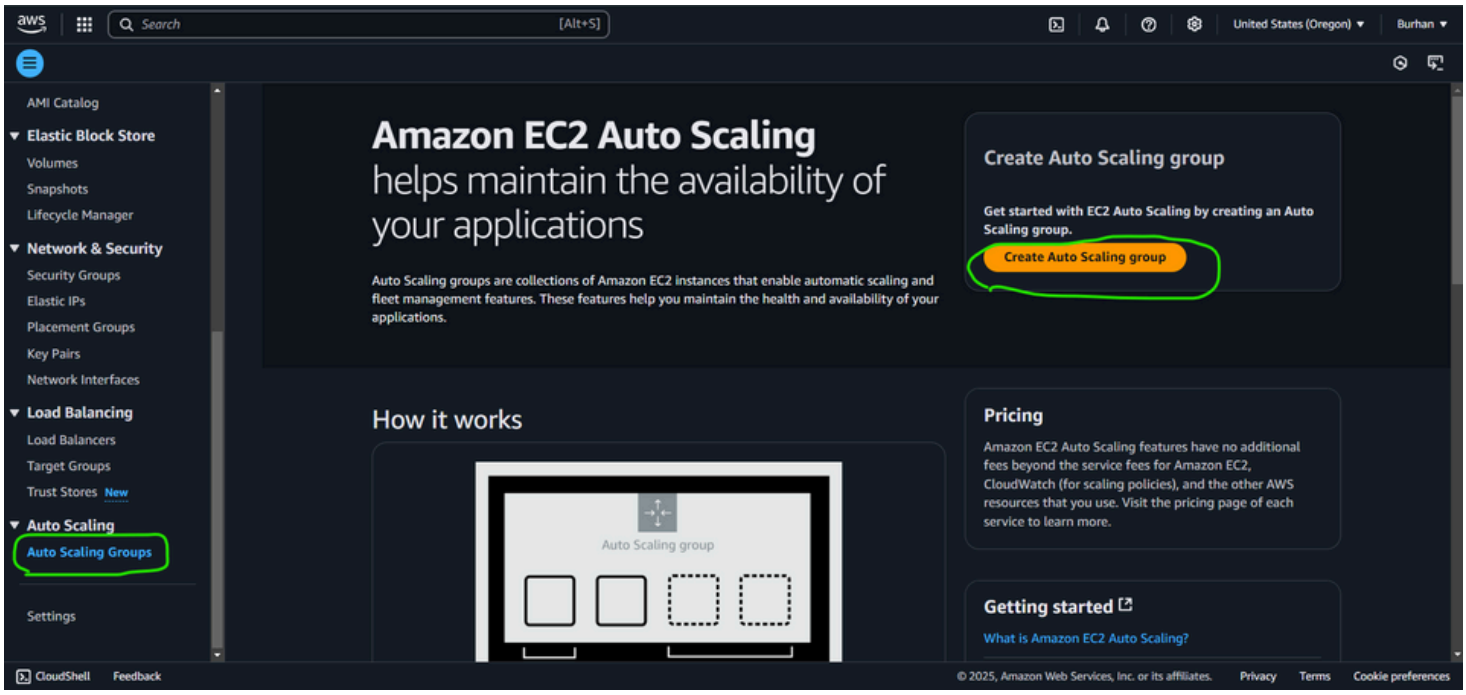
In short, ASG helps your application handle traffic changes automatically while saving money!

Auto Scaling Group Hands On

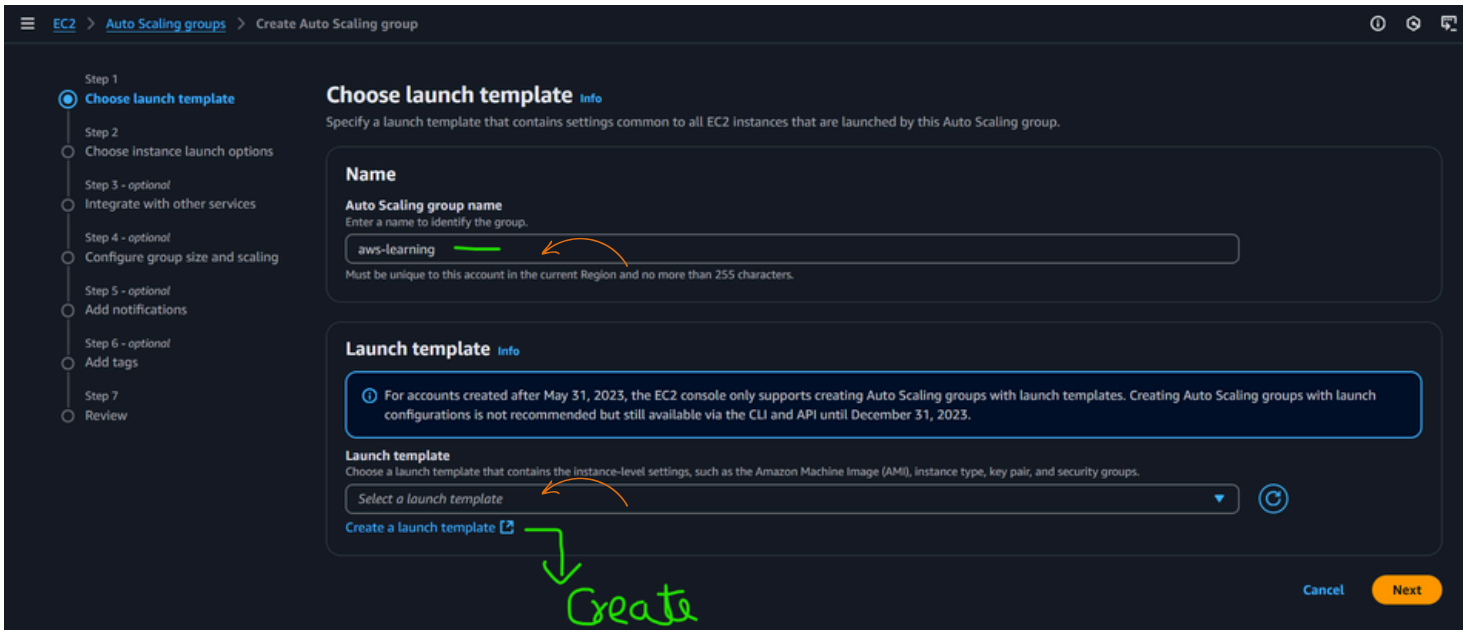
Remove All old Configuration



Create ASG Auto Scaling Groups



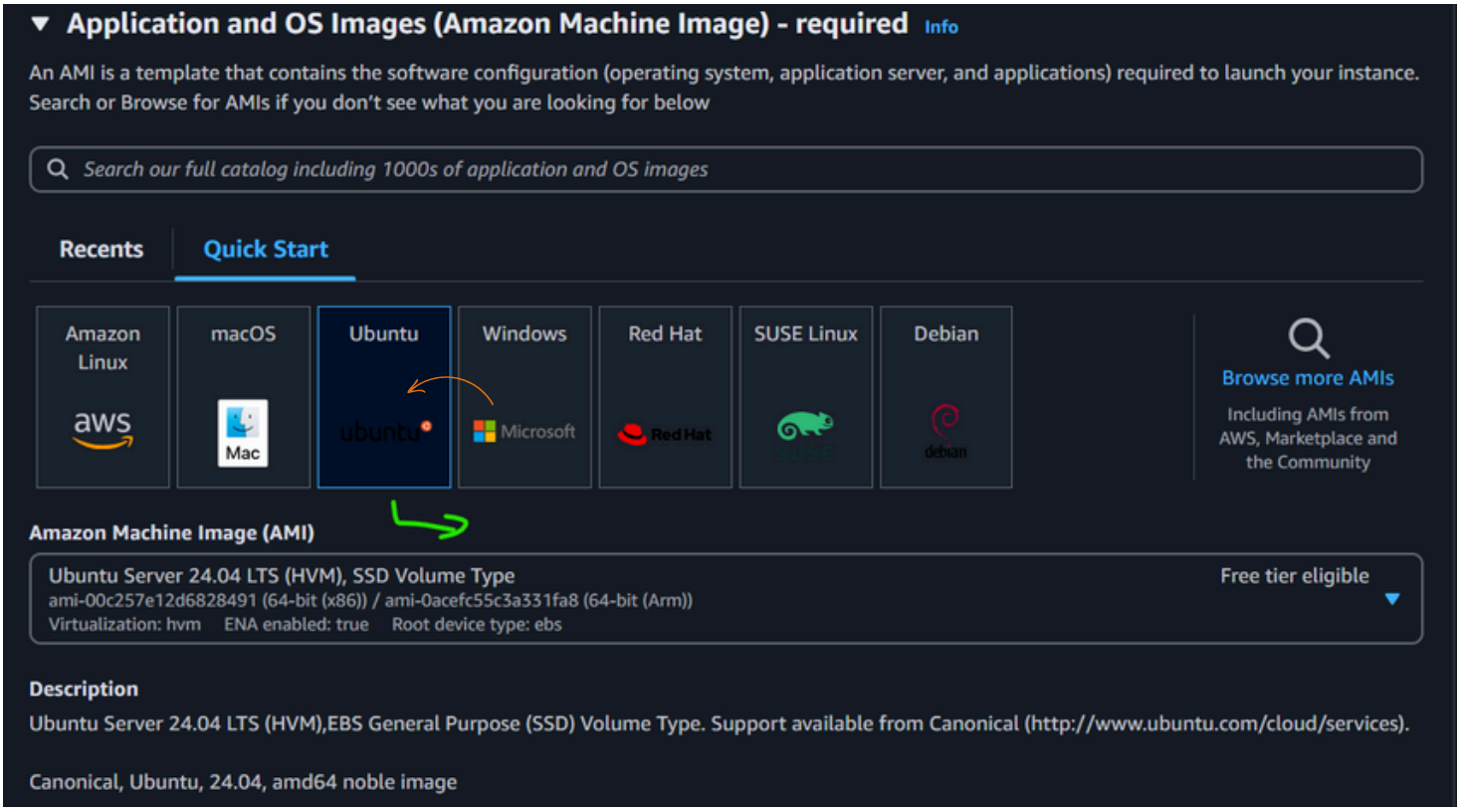
Give name ASG : aws-learning



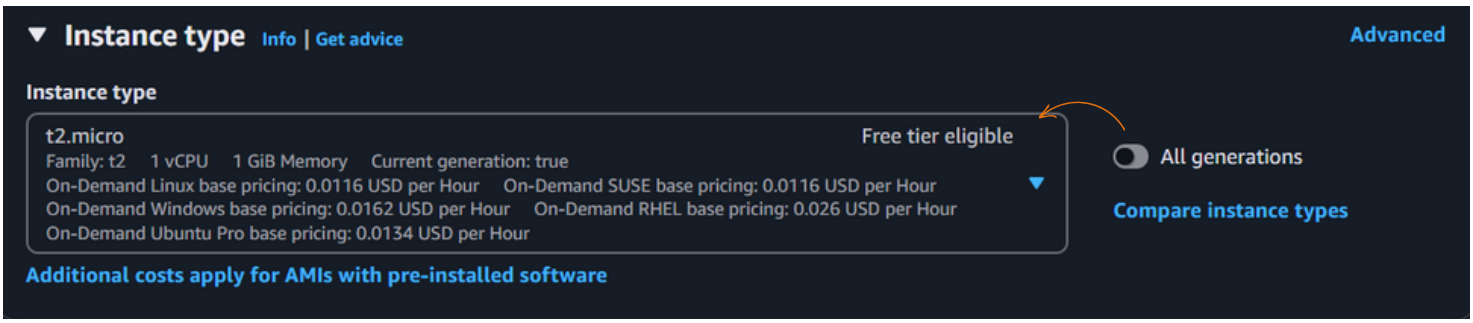
Create a launch template

This template is begin used to tell to the Auto Scaling Group How to Create EC2 within it

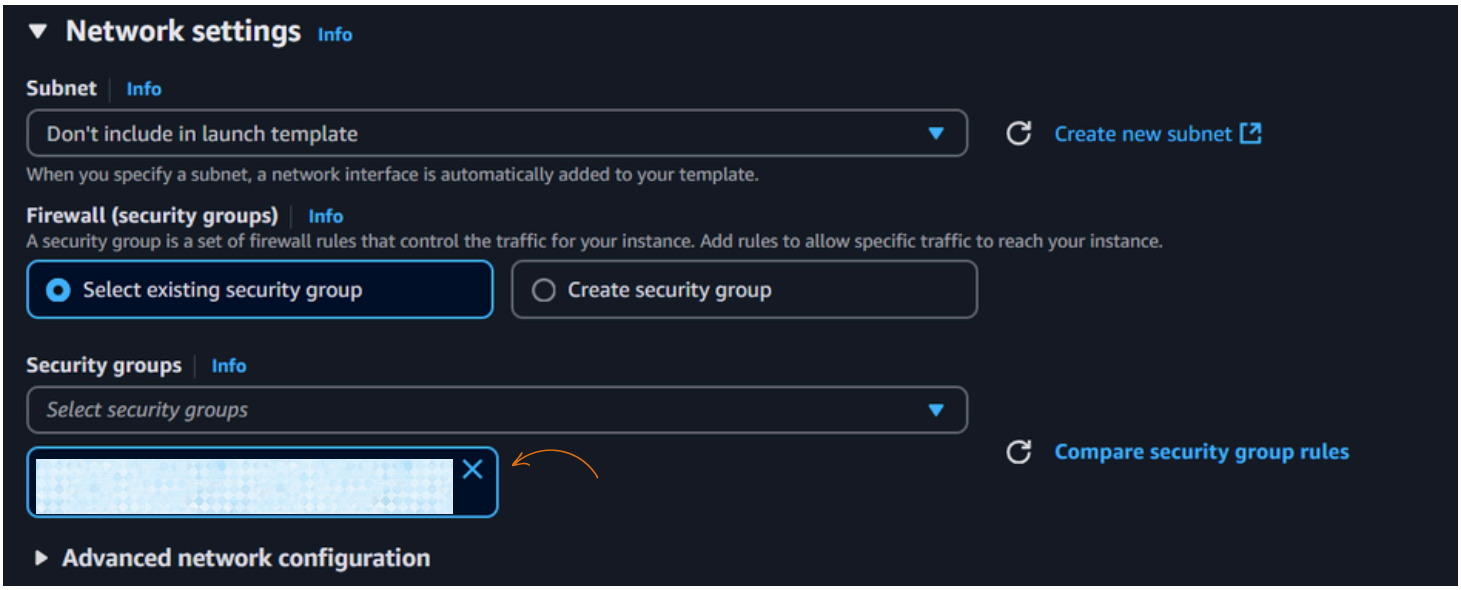
Create ASG Auto Scaling Groups



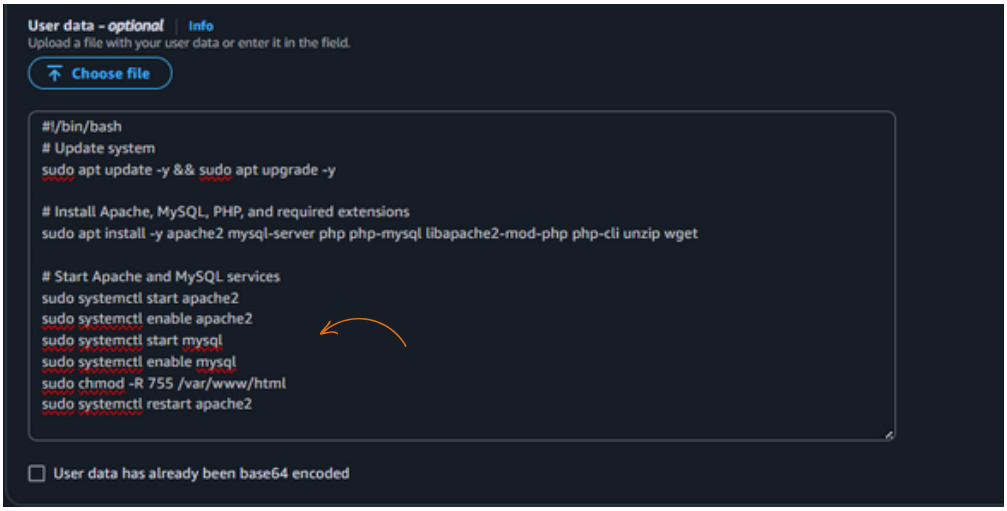
Instance type



Network settings

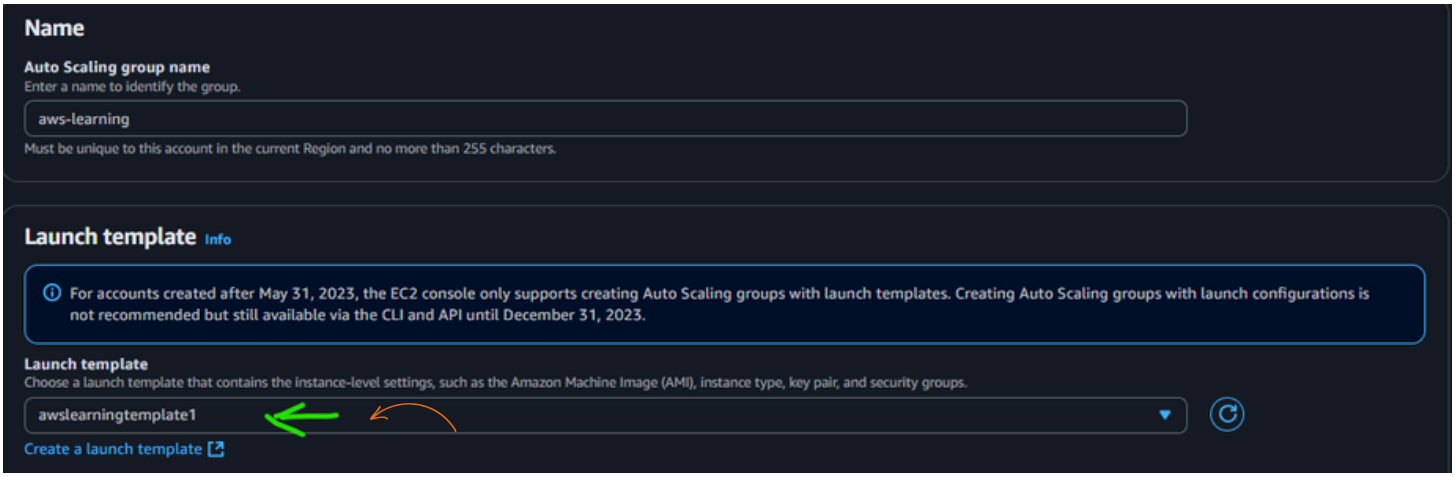


Advance > User data

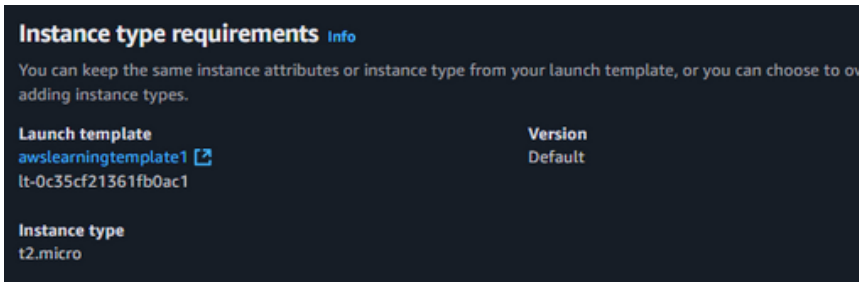



```
#!/bin/bash
# Update system
sudo apt update -y && sudo apt upgrade -y

# Install Apache, MySQL, PHP, and required extensions
sudo apt install -y apache2 mysql-server php php-mysql libapache2-mod-php php-cli unzip wget

# Start Apache and MySQL services
sudo systemctl start apache2
sudo systemctl enable apache2
sudo systemctl start mysql
sudo systemctl enable mysql
sudo chmod -R 755 /var/www/html
sudo systemctl restart apache2
```

Refresh Launch Template

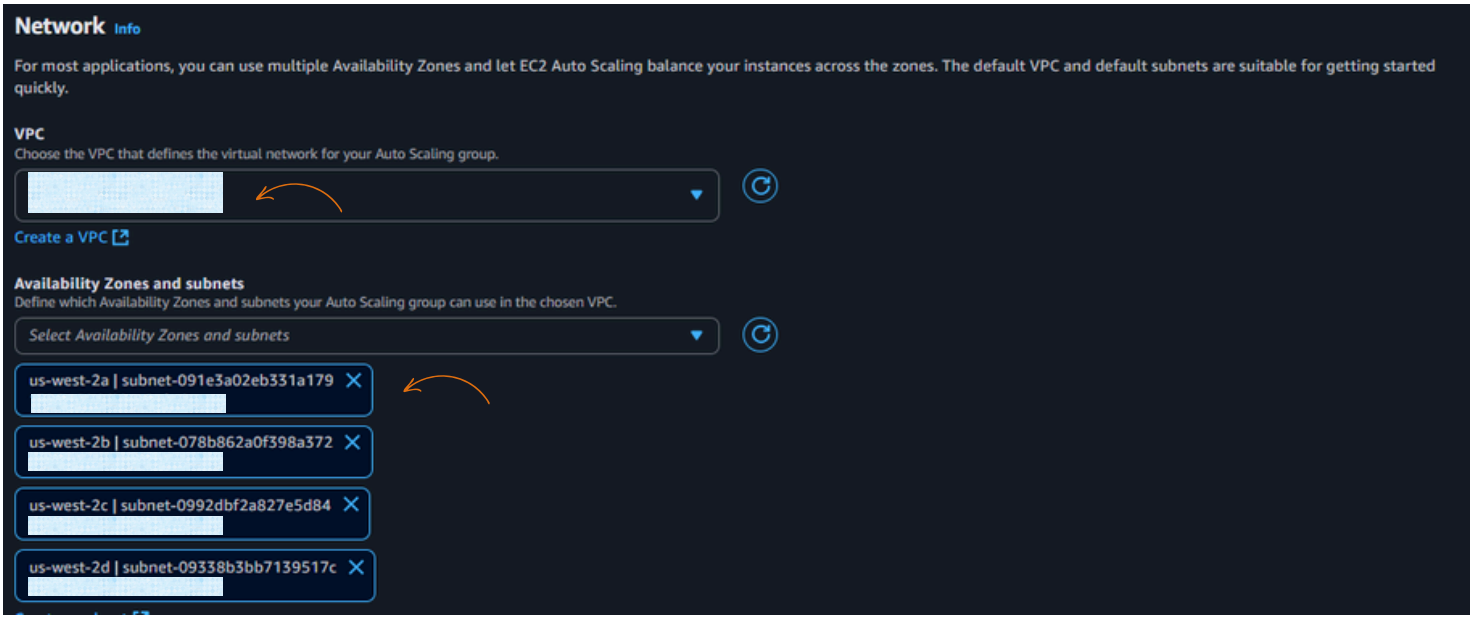


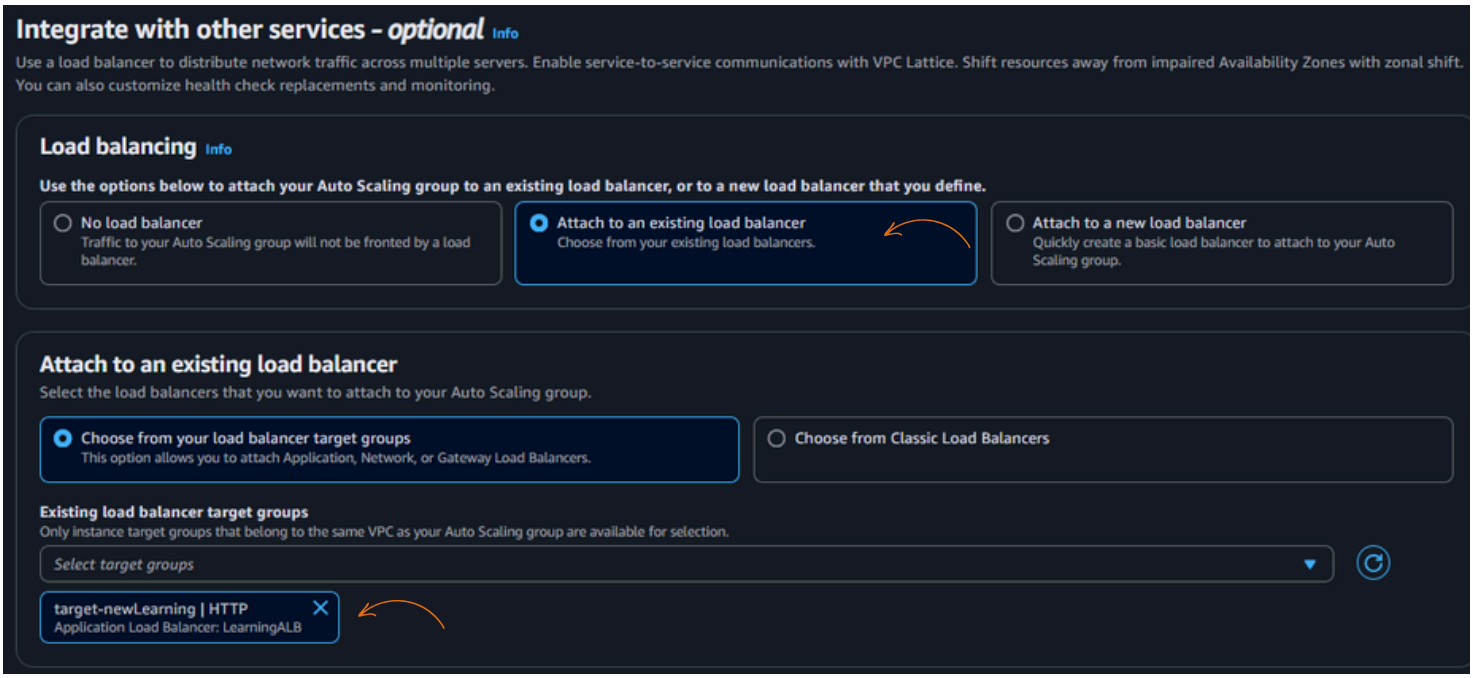
Click On Next

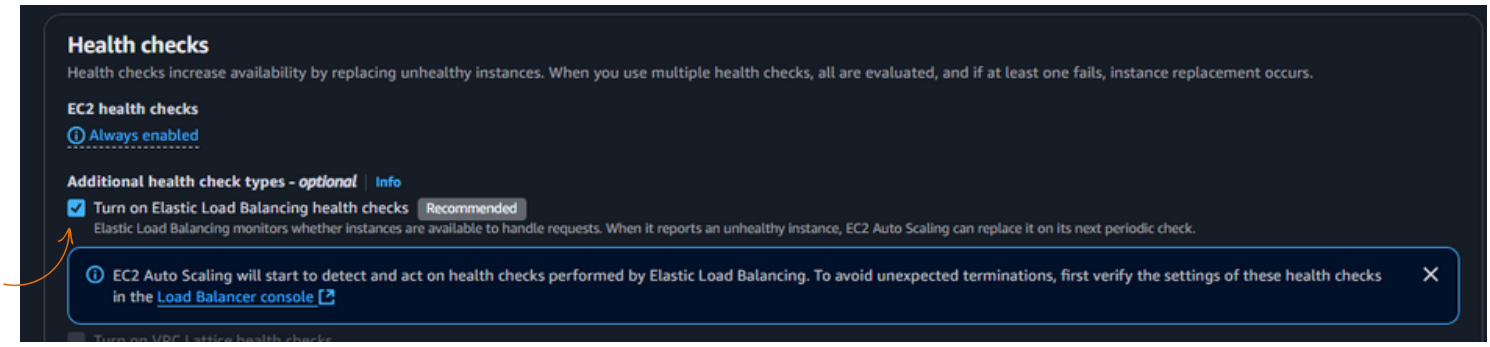


Select Multi Availability **Zone**
click on **Next**

Advance > User data

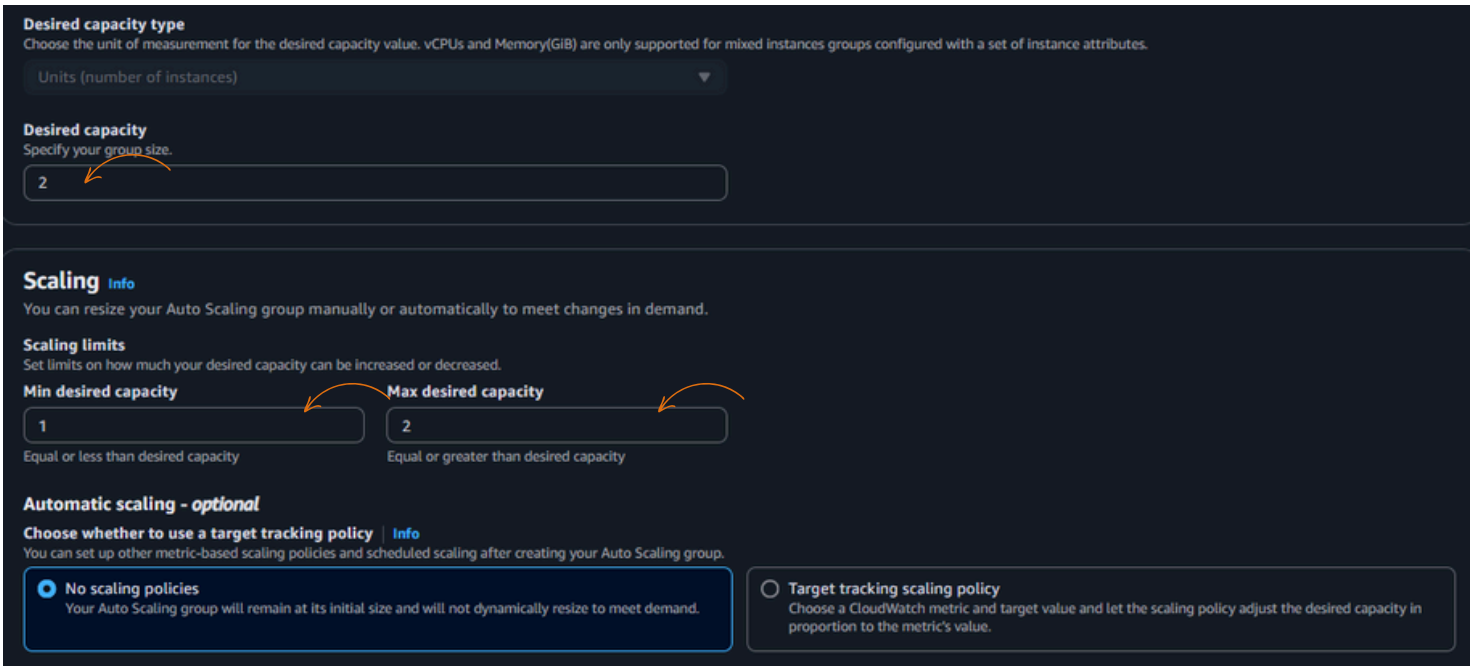




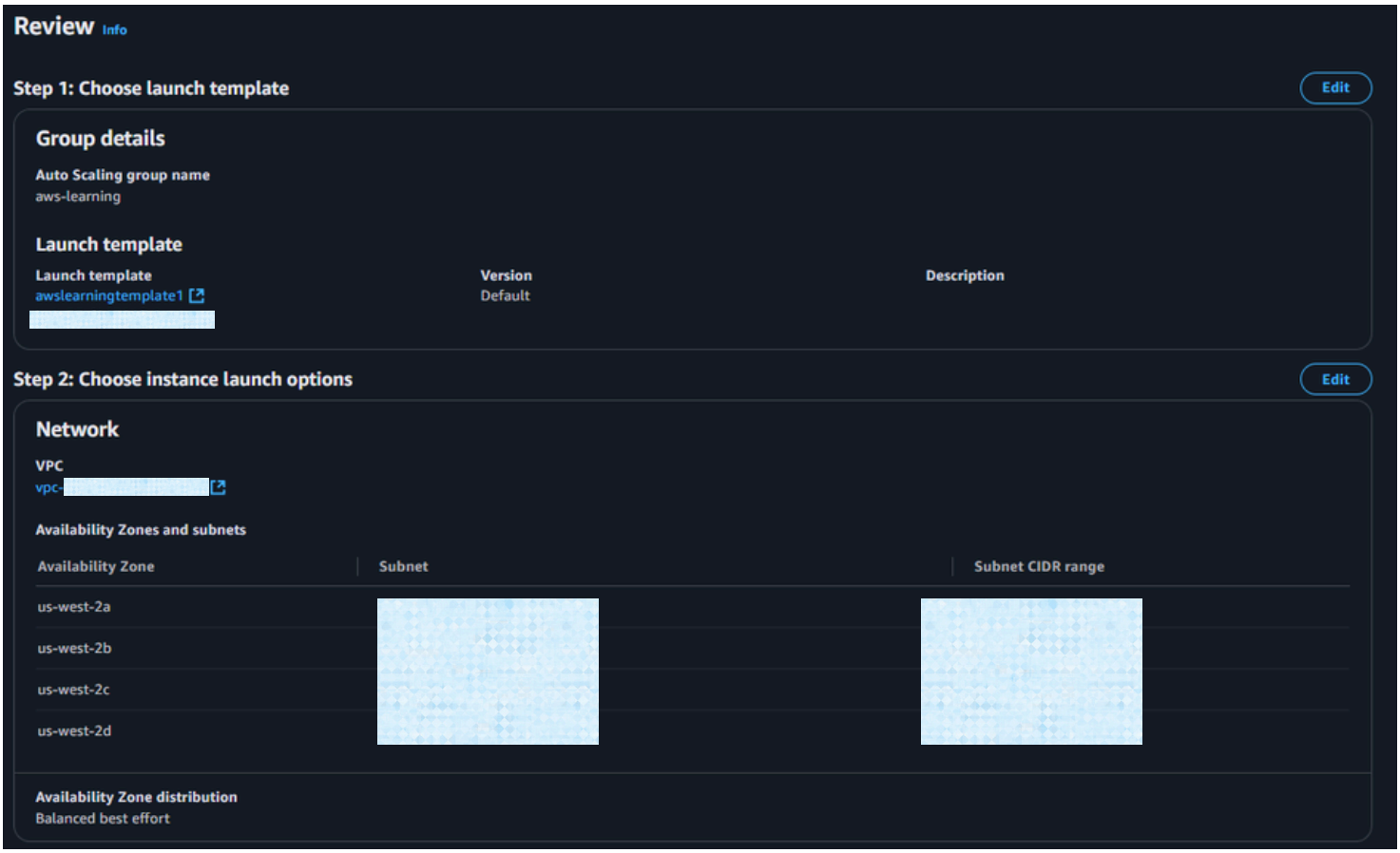


Click On Next

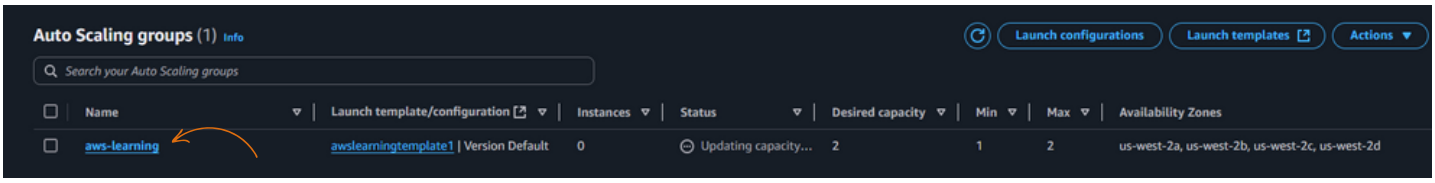
Desired capacity



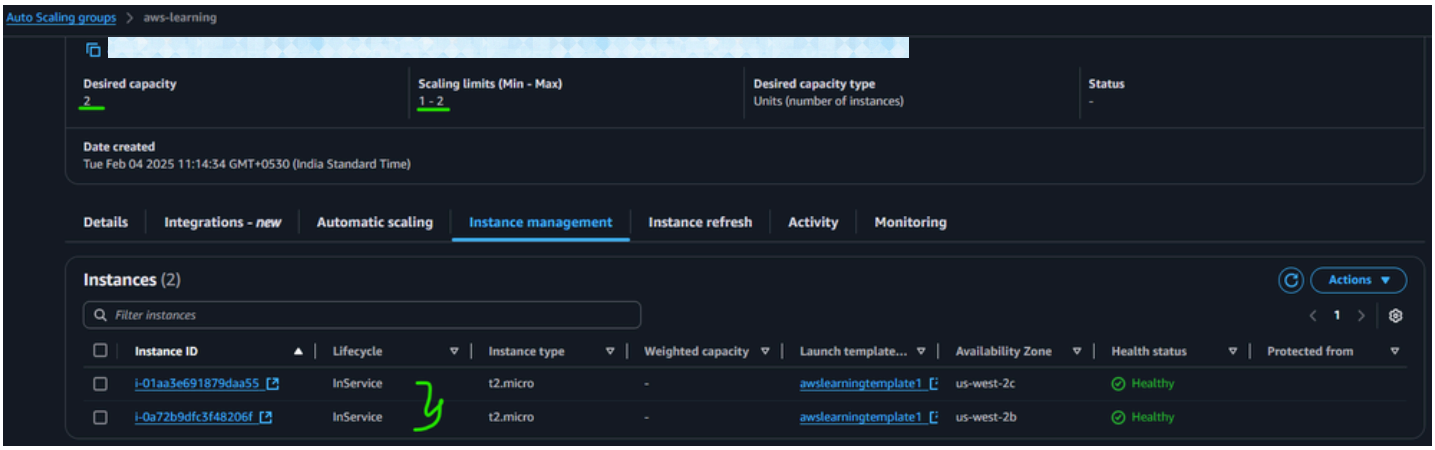
Review Click On Next



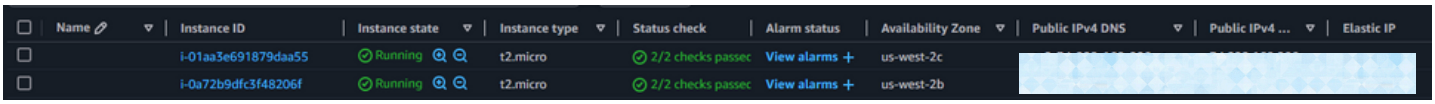
Desired capacity



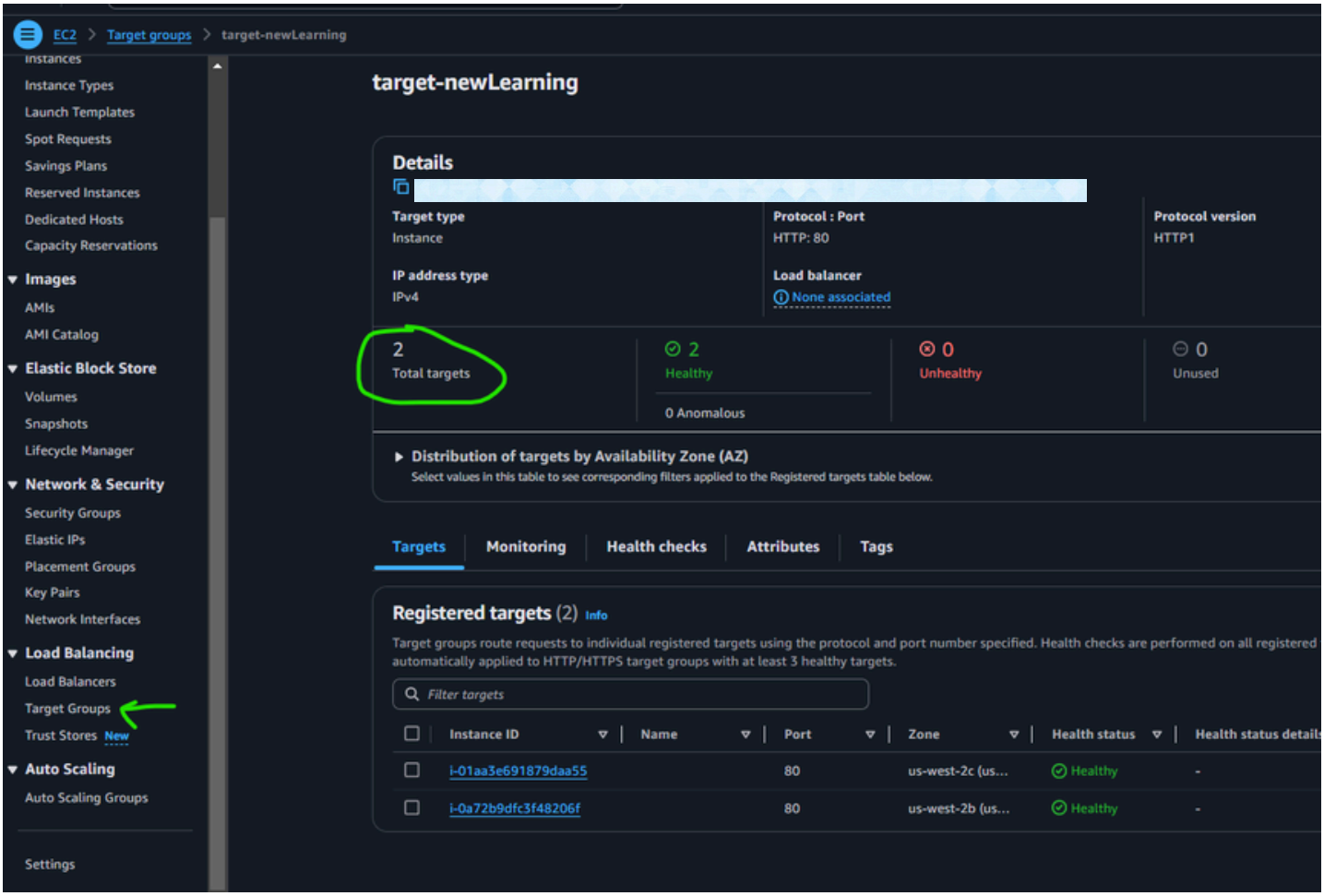
Instance management



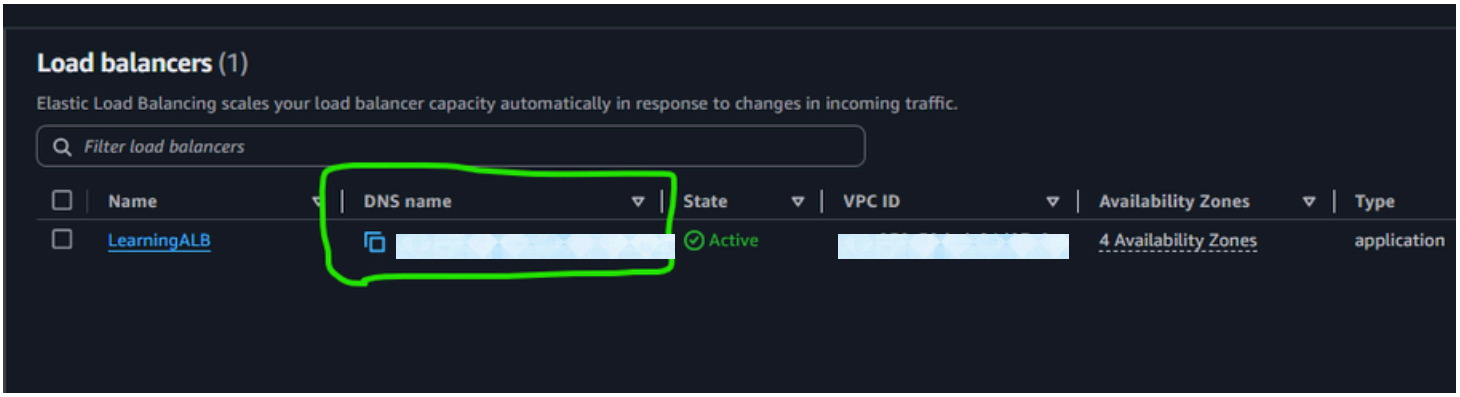
Tow Instance are running this instance are created by ASG : auto scaling group



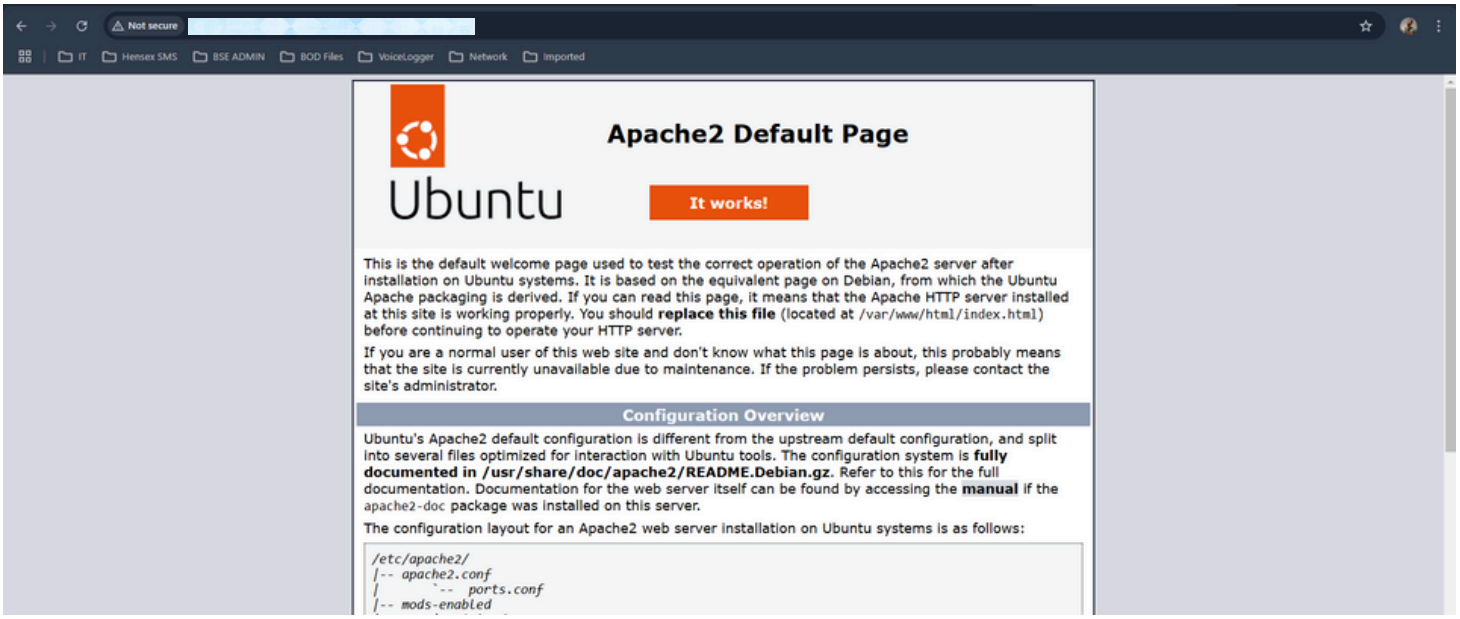
Target group



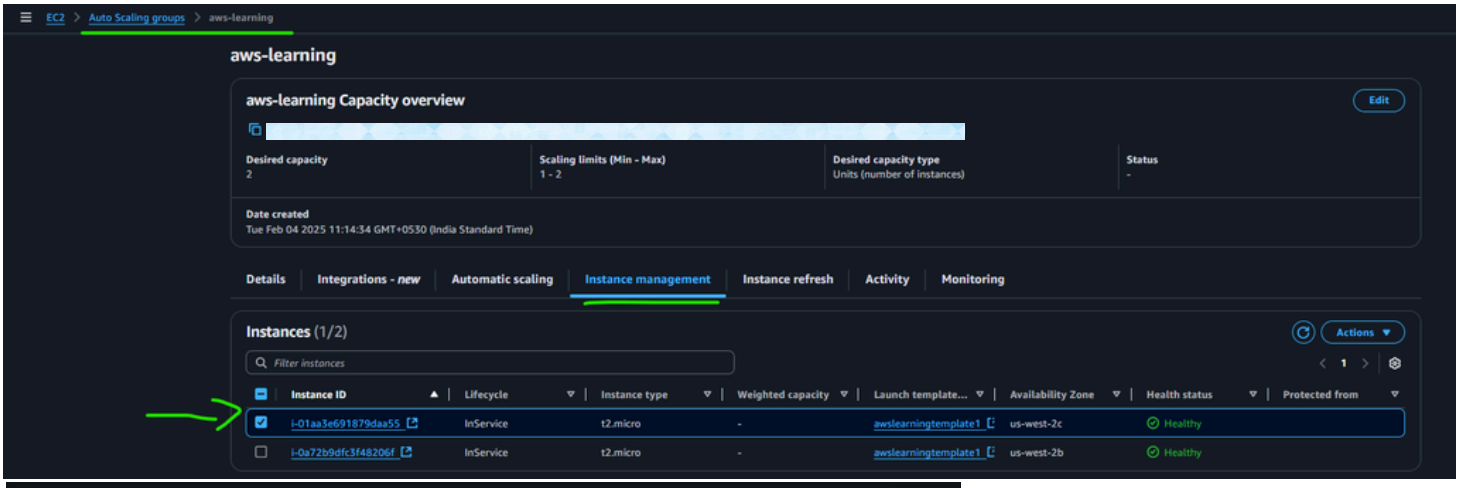
Access From Load Balancer

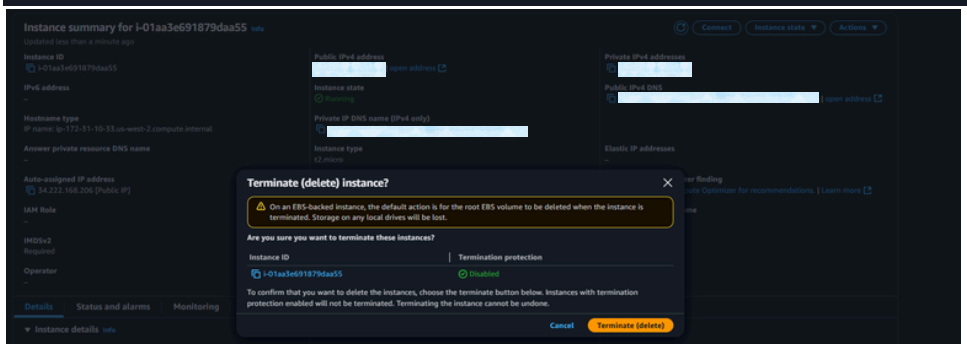


Paste on Browser



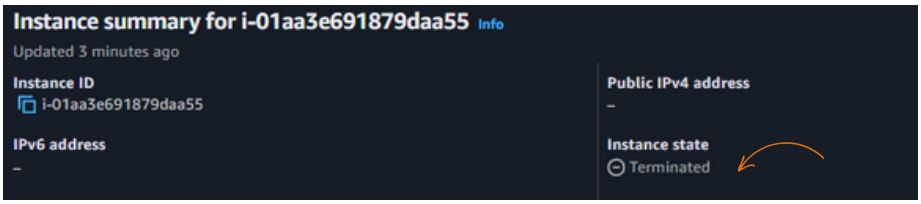
Webpage access form both instance we can Terminated 1 running instance



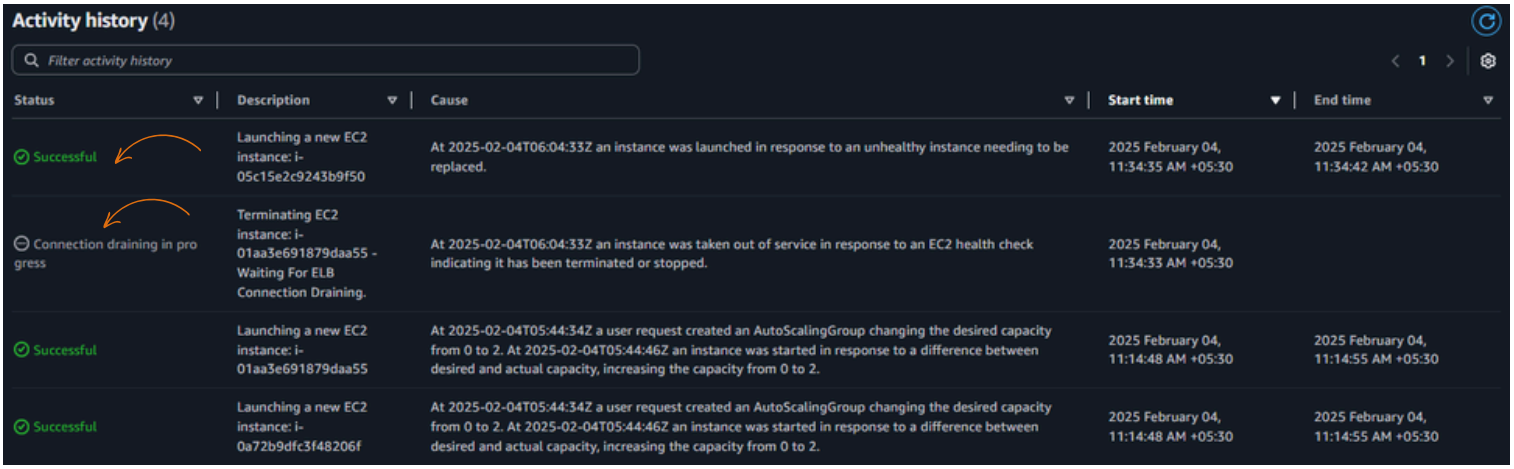


Desired capacity

We have set a minimum of 2 and a desired capacity in the Auto Scaling Group. If one instance gets terminated, the Auto Scaling Group automatically creates a new instance



Power of ASG

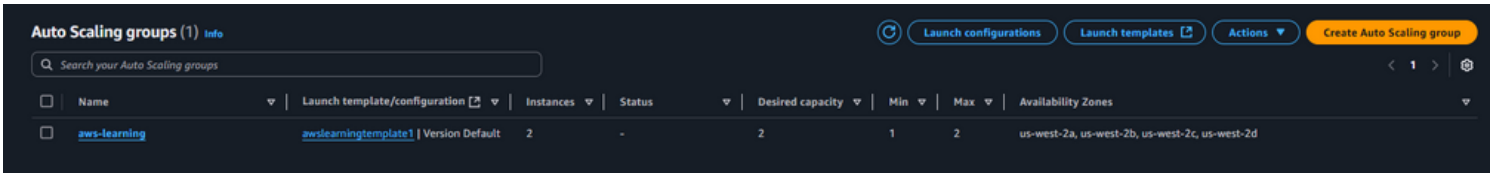


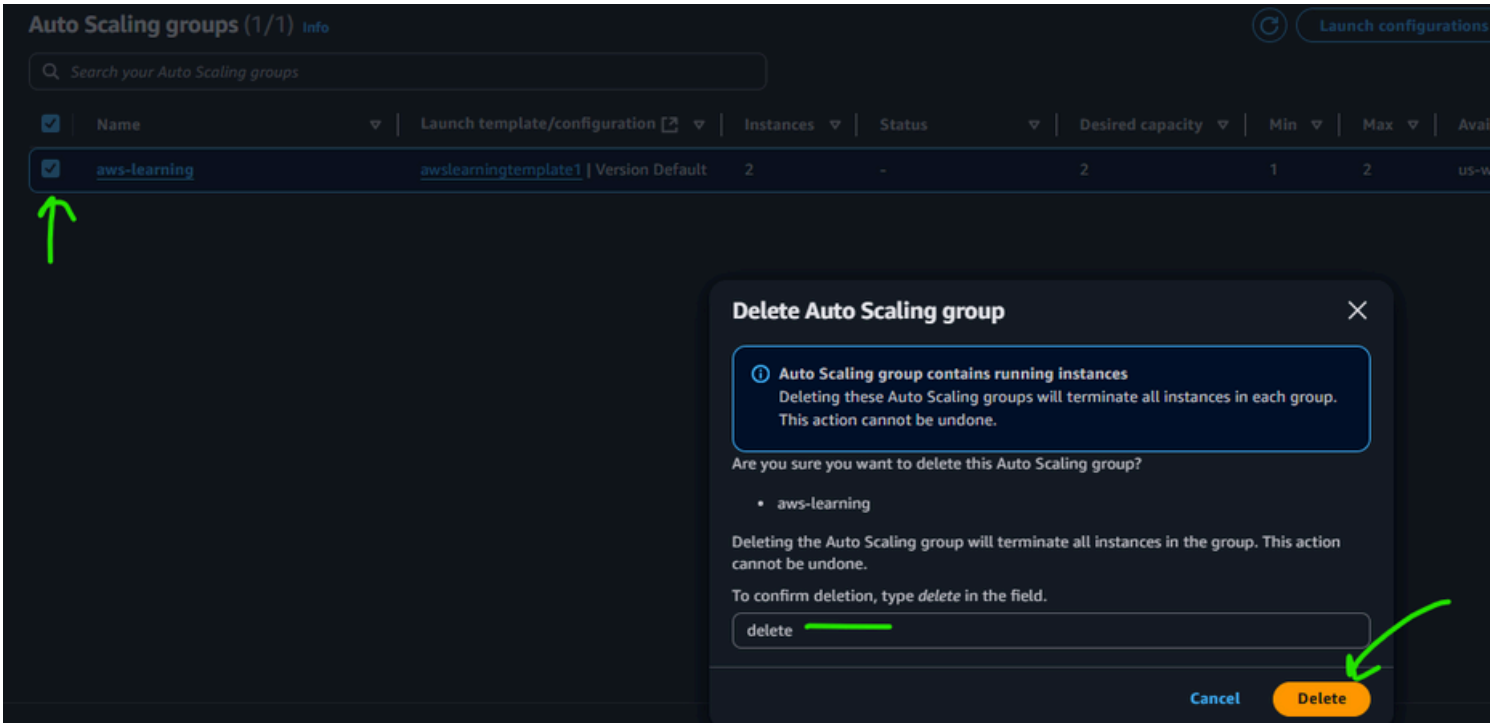
Auto Scaling Group (ASG) in AWS automatically detects unhealthy instances and replaces them with new ones

Clean Up All Configuration

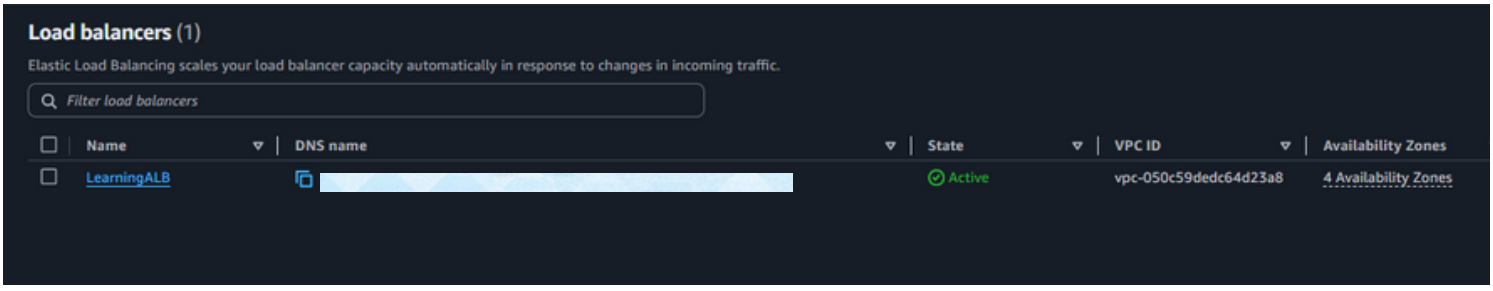
If you manually delete an instance that is part of an Auto Scaling Group (ASG), the ASG will automatically launch a new instance to maintain the desired capacity

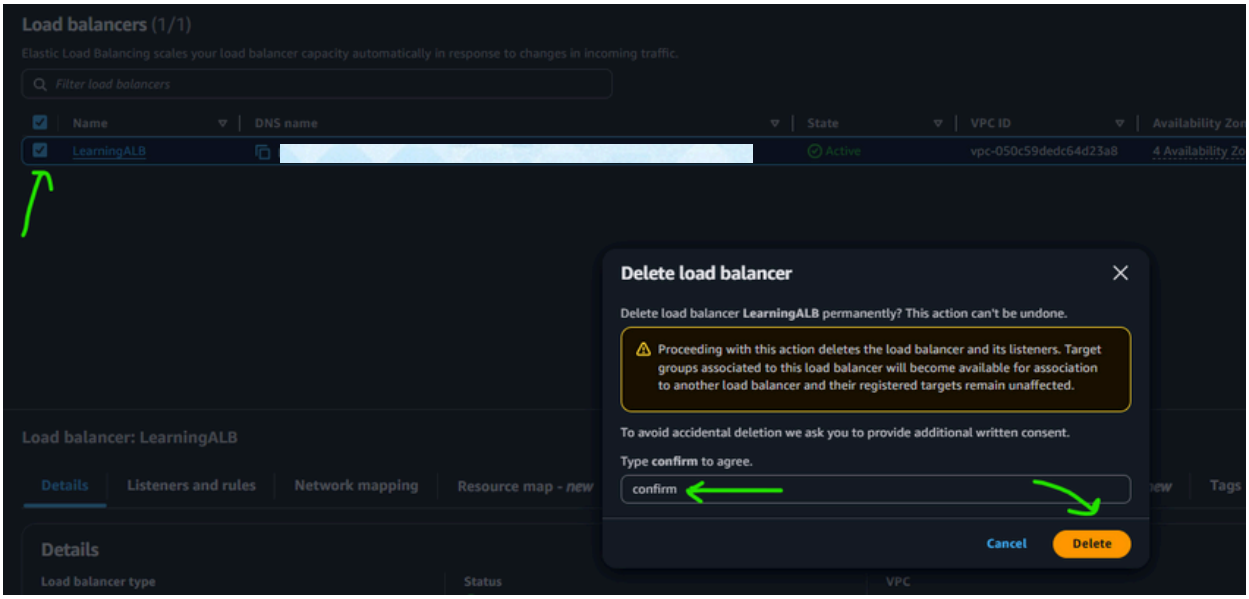
Click on Auto Scaling Group





Delete Load Balancer





Should i delete my target group ?
it not cost your any money and the TG is going to empty because we are delete Auto Scaling Group

When Your ASG is Deleted the your EC2 Instacne auto deleted