# Little Bit Advance Labs

## Part 1: EC2 with ELB and ASG

**Objective**: Learn how to create a scalable and highly available web application environment using Amazon EC2 instances, ELB, and ASG.

**Approach**:

1. **Launch EC2 Instances**: Start by launching two or more EC2 instances. These instances will run a simple web application (e.g., a "Hello World" page or any basic web service).
2. **Configure Load Balancer**: Set up an Elastic Load Balancer (ELB) to distribute incoming web traffic across your EC2 instances. This step ensures high availability and fault tolerance.
3. **Set Up Auto Scaling Group (ASG)**: Create an ASG that uses the launched EC2 instances. Configure ASG policies to automatically scale the number of instances up or down based on criteria like CPU usage or network traffic.
4. **Test Your Setup**: Simulate traffic to test the scaling policies and the load balancer. Observe how ASG adds or removes instances and how ELB distributes traffic.
5. **Verify Website Functionality**: Ensure that the website hosted on EC2 instances remains accessible and functional during scaling operations.

**Goal**: By the end of this lab, students will have a hands-on understanding of setting up a load-balanced and auto-scaled web application using AWS services.

**SOLUTION:**

**Launch EC2 Instances:**

- Log in to your AWS Management Console.
- Navigate to the EC2 dashboard.
- Click on "Launch Instance" to start the instance creation process.
- Choose an Amazon Machine Image (AMI) based on your requirements (e.g., Amazon Linux, Ubuntu, etc.).
- Select an instance type (e.g., t2.micro for testing purposes).
- Configure instance details such as network settings, subnet, IAM role, etc.
- Add storage as needed.
- Configure security group to allow HTTP/HTTPS traffic.
- Review and launch the instance.

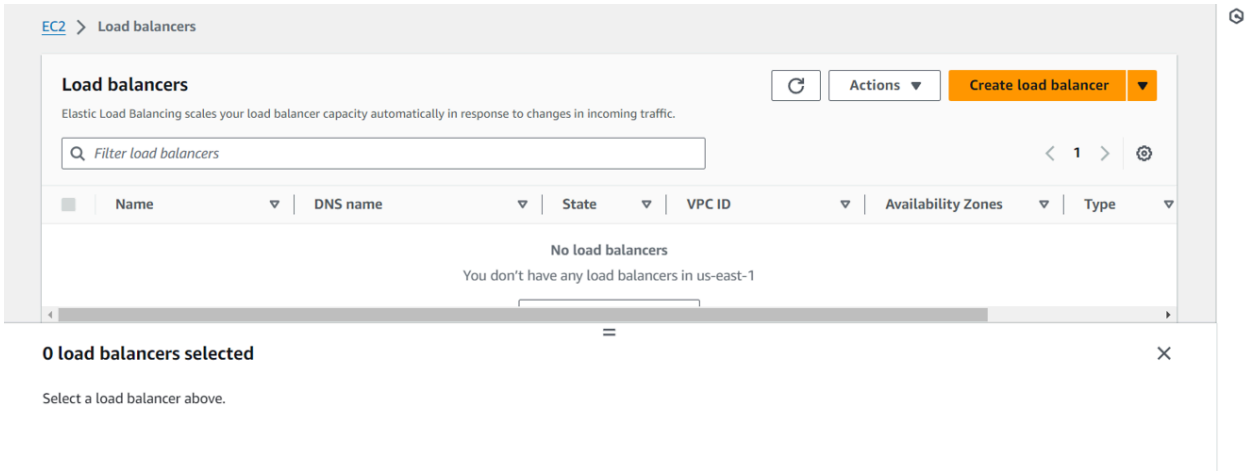
- Repeat the above steps to launch at least two EC2 instances.

  Done in basic labs so using a server from the previous lab.

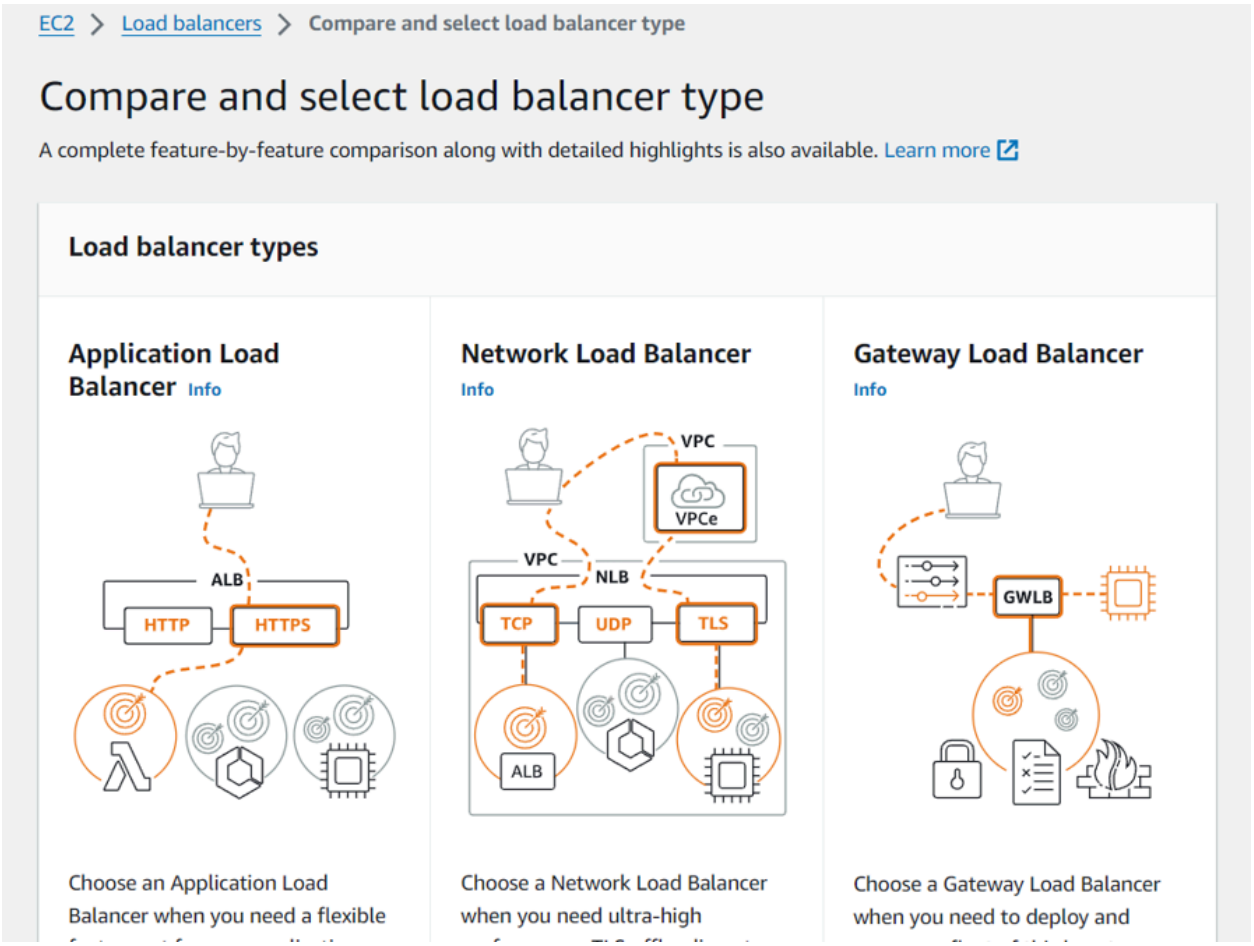
**After that we need to create load balancer**

**Configure Load Balancer:**

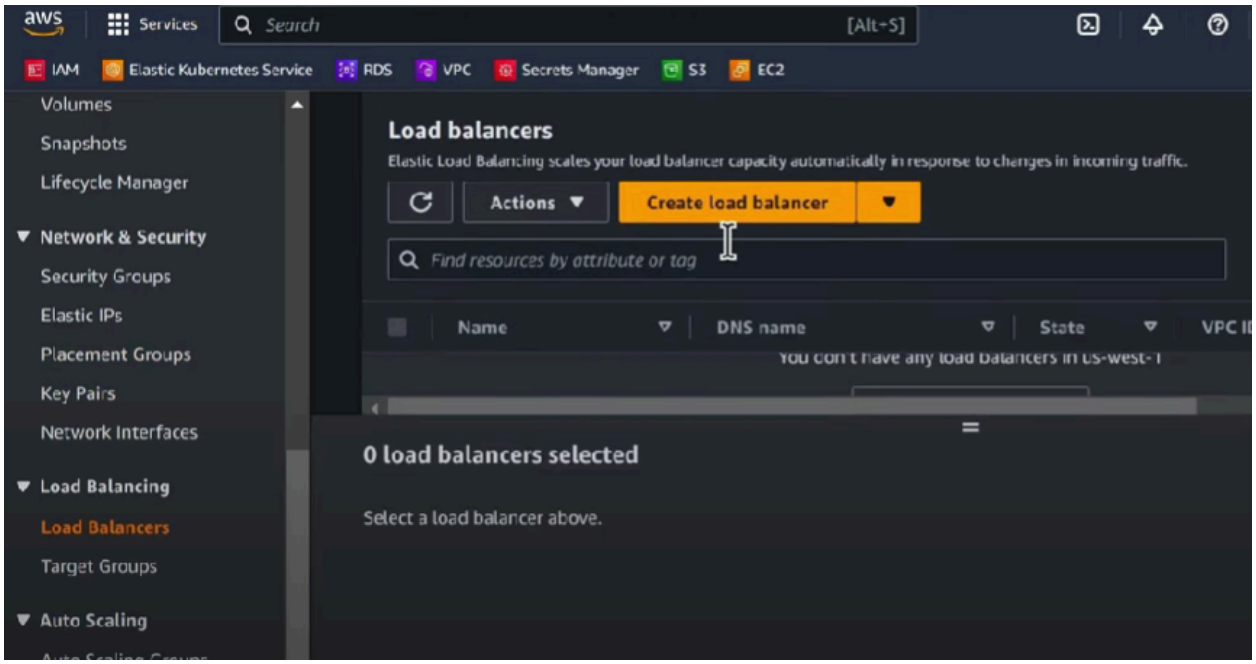
- Go to the EC2 dashboard and click on "Load Balancers" in the navigation pane.

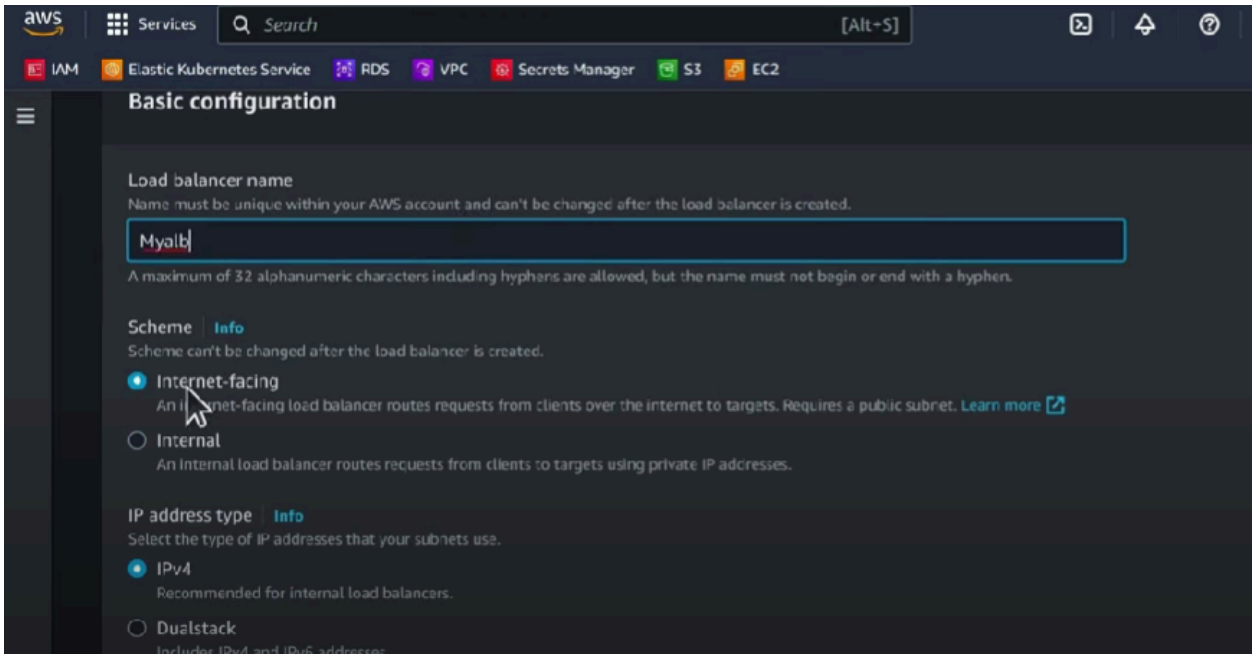


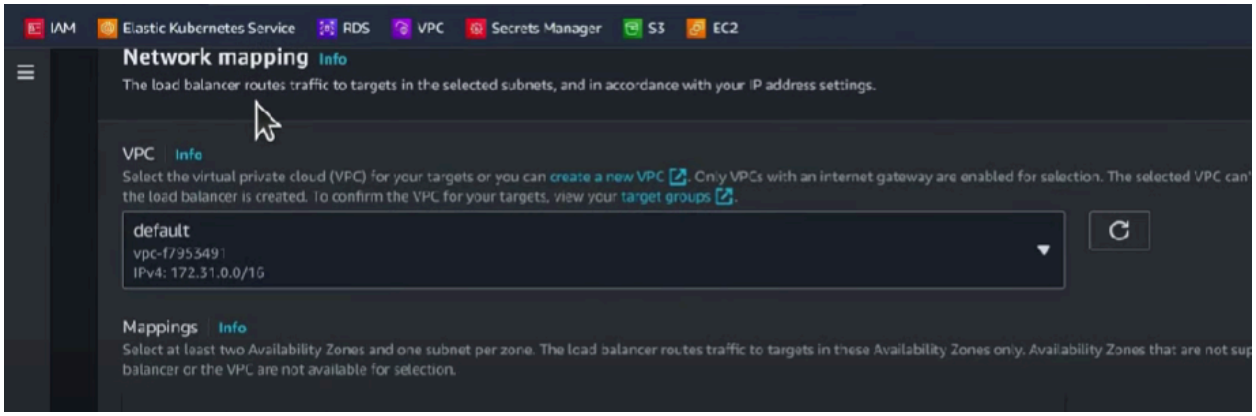

- Click on "Create Load Balancer" and select the type of load balancer (usually Application Load Balancer is preferred).



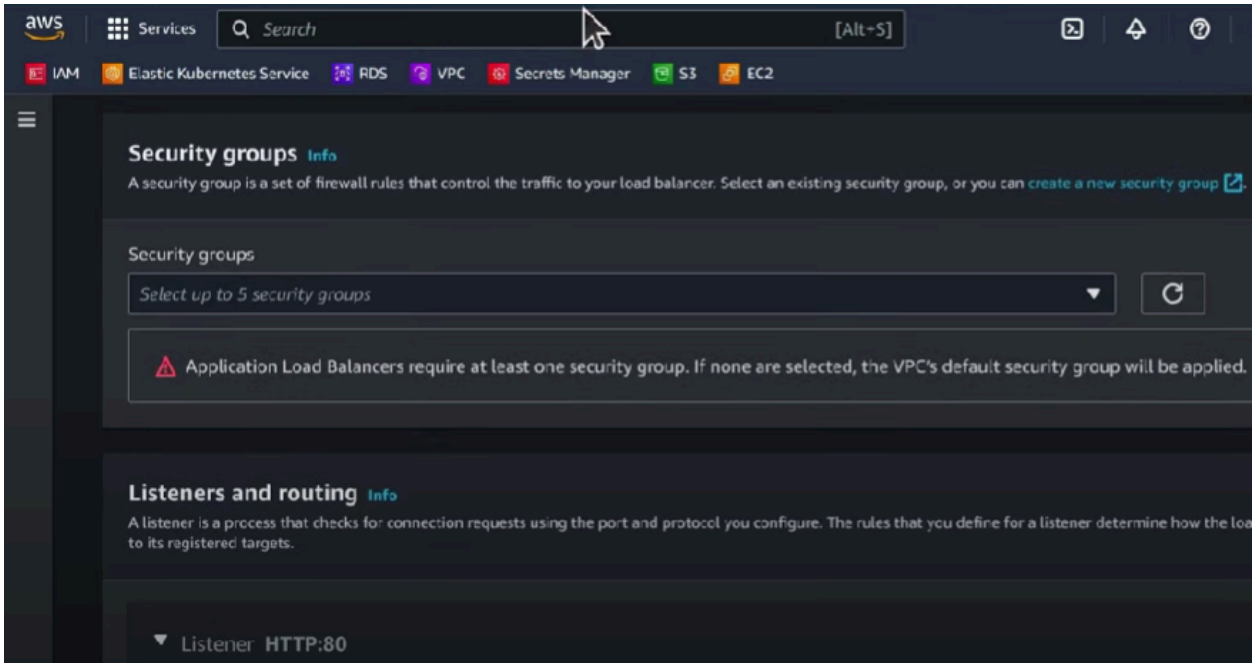

- Configure load balancer settings including name, listener configuration (e.gHTTP or HTTPS), VPC, available subnets, availability zones, and security settings.
- Click on "Create Load Balancer" and select the type of load balancer (usually Application Load Balancer is preferred)

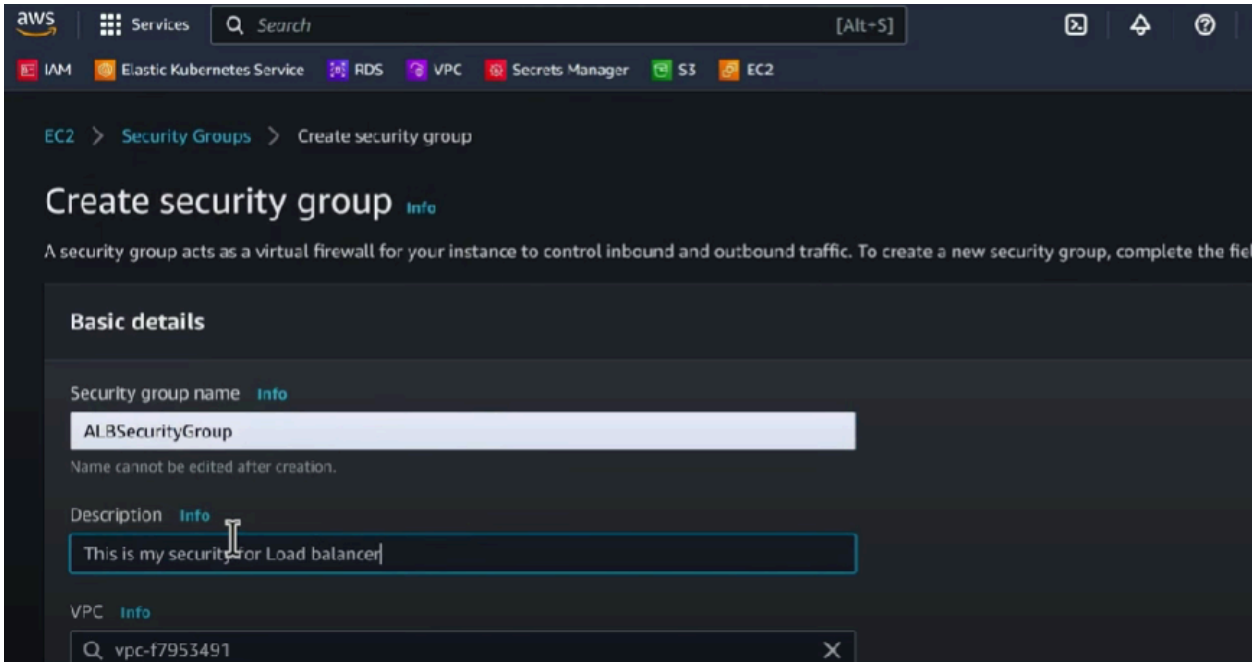

aws
Services
Search
[Alt+S]
IAM
Elastic Kubernetes Service
RDS
VPC
Secrets Manager
S3
EC2
Volumes
Snapshots
Lifecycle Manager
Network & Security
Security Groups
Elastic IPs
Placement Groups
Key Pairs
Network Interfaces
Load Balancing
Load Balancers
Target Groups
Auto Scaling
Load balancers
Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.
Actions
Create load balancer
Find resources by attribute or tag
Name
DNS name
State
0 load balancers selected
Select a load balancer above.

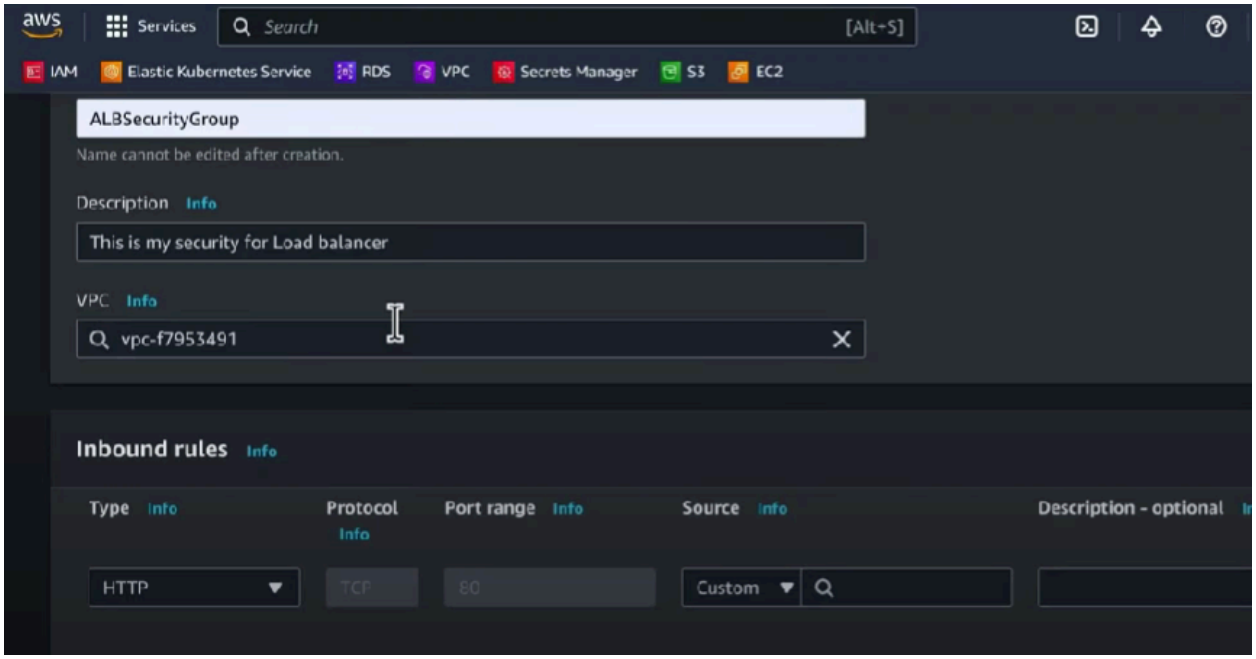

aws
Services
Search
[Alt+S]
IAM
Elastic Kubernetes Service
RDS
VPC
Secrets Manager
S3
EC2
Basic configuration
Load balancer name
Name must be unique within your AWS account and can't be changed after the load balancer is created.
Myalb
A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.
Scheme
Info
Scheme can't be changed after the load balancer is created.
Internet-facing
An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet.
Learn more
Internal
An internal load balancer routes requests from clients to targets using private IP addresses.
IP address type
Info
Select the type of IP addresses that your subnets use.
IPv4
Recommended for internal load balancers.
Dualstack



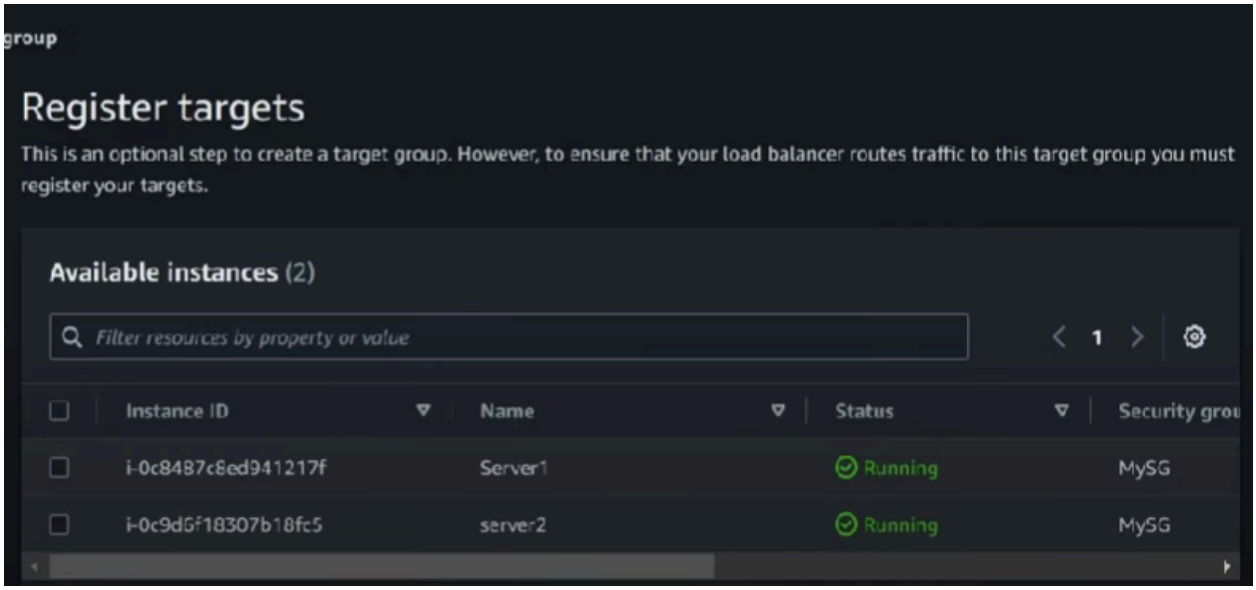

- Now  we need security group, either we use selected available security groups or create new security group
-
- Now  we need security group, either we use selected available security groups or create new security group



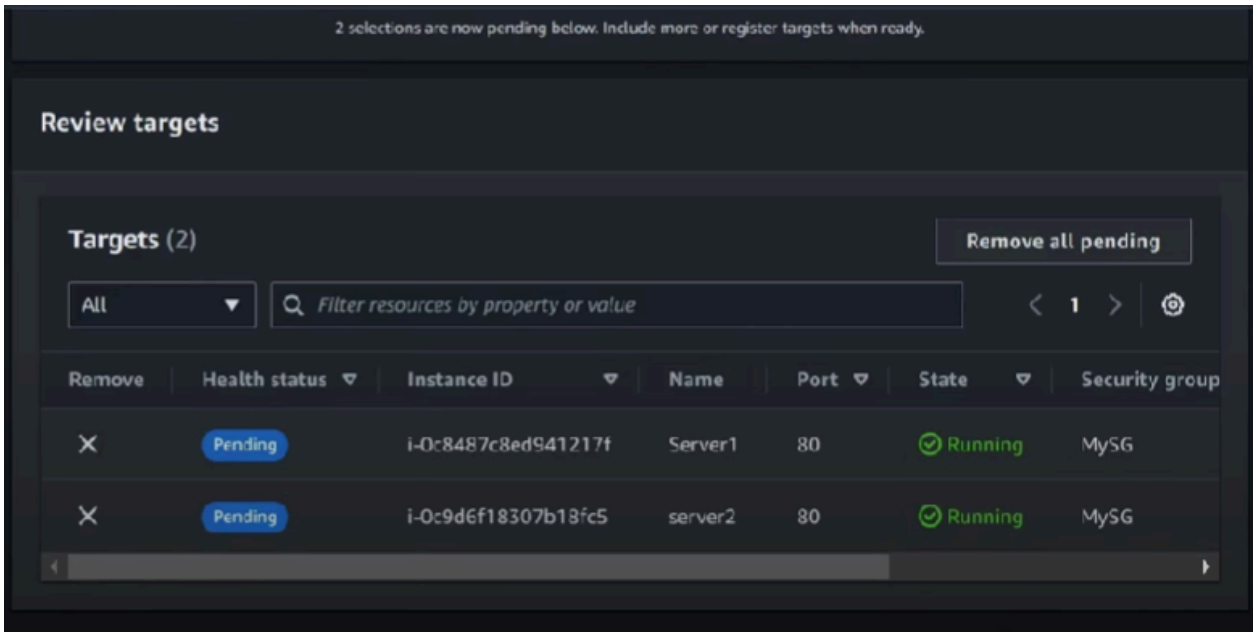


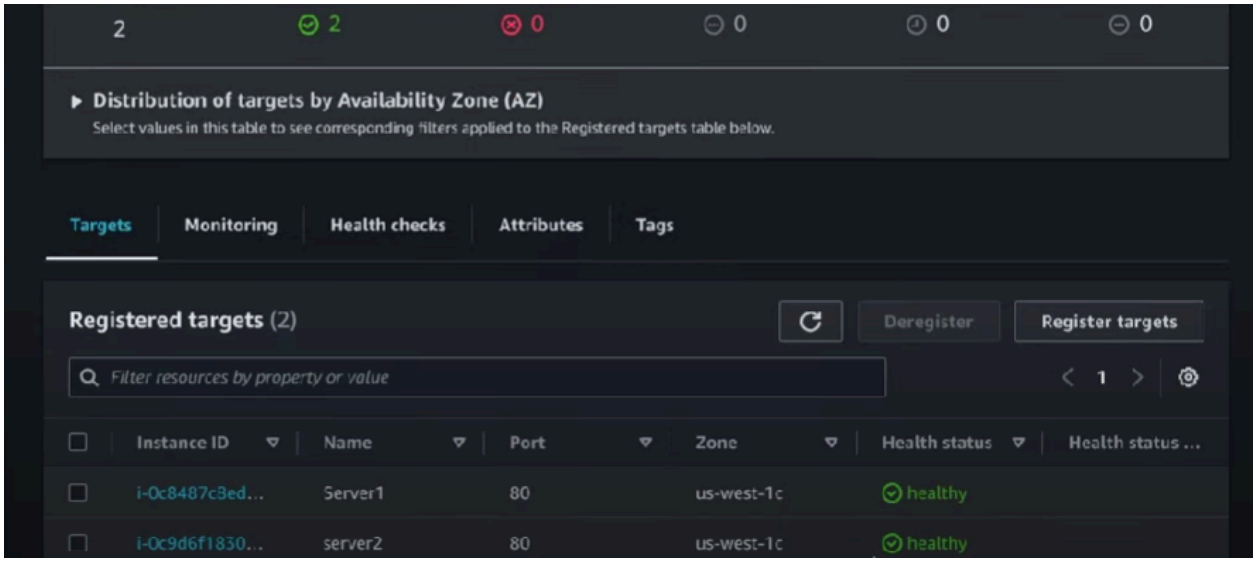

- We also need a target group , so “create target group”




- We need to select instances from the list of available instances and click on include as pending below and then click on the “target group” button.

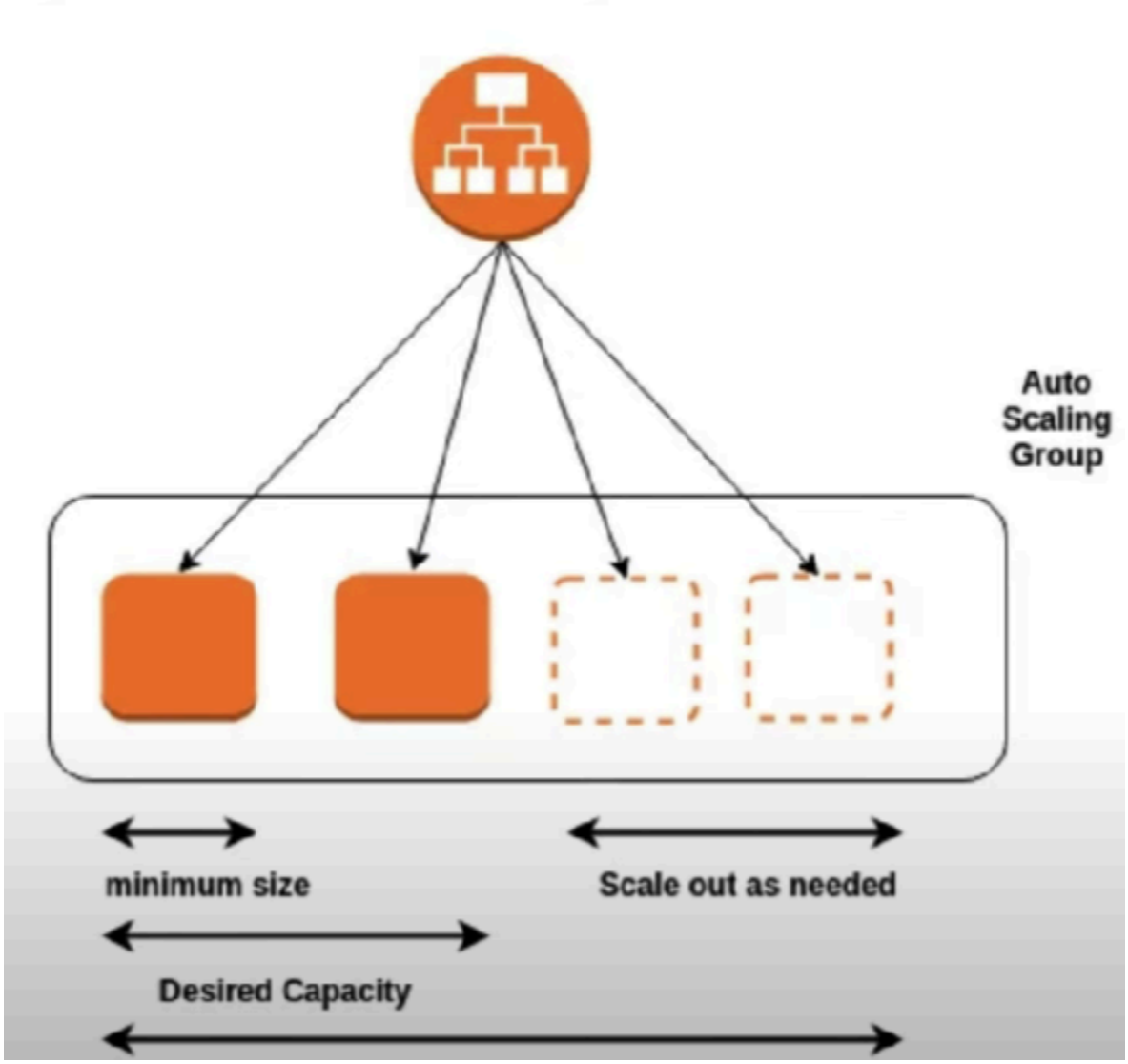



- Then our load balancer will be created and it will be available which we can select and then click on the “create load balancer” button.
- In the target group we need to check if all our instances are healthy, if they are not healthy we will not be able to see the traffic.




- To check if the server is running, copy the DNS name from the newly created load balancer and paste in the browser.

**AUTO SCALING GROUP:**

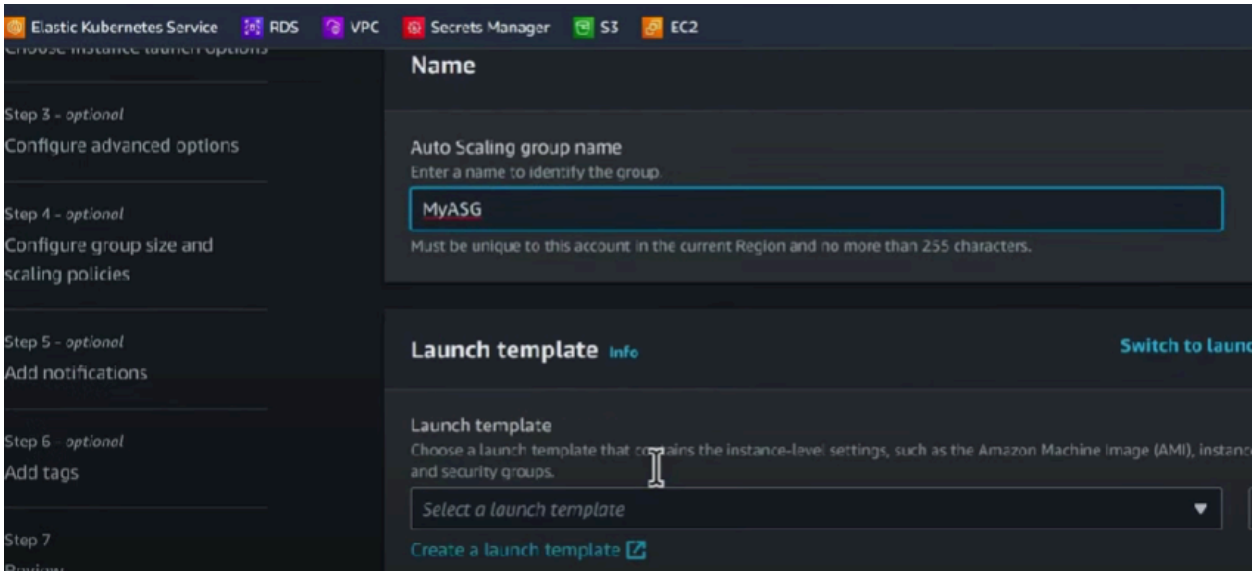


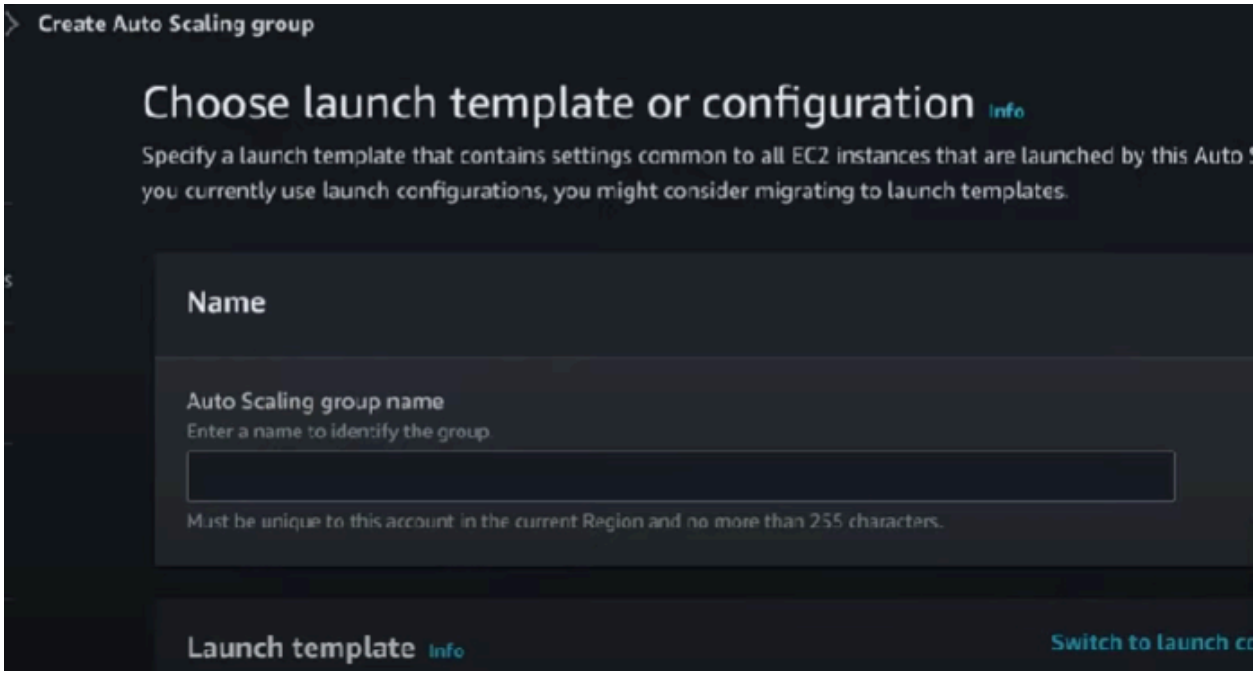

Here we have created 2 servers manually but if traffic is increased, we might want to have more servers and in those situations the server should be increased and if less traffic then server should be decreased.

For this purpose we need an Auto Scaling Group which can automatically create the required number of servers depending on the traffic and also delete the server if the traffic is less.

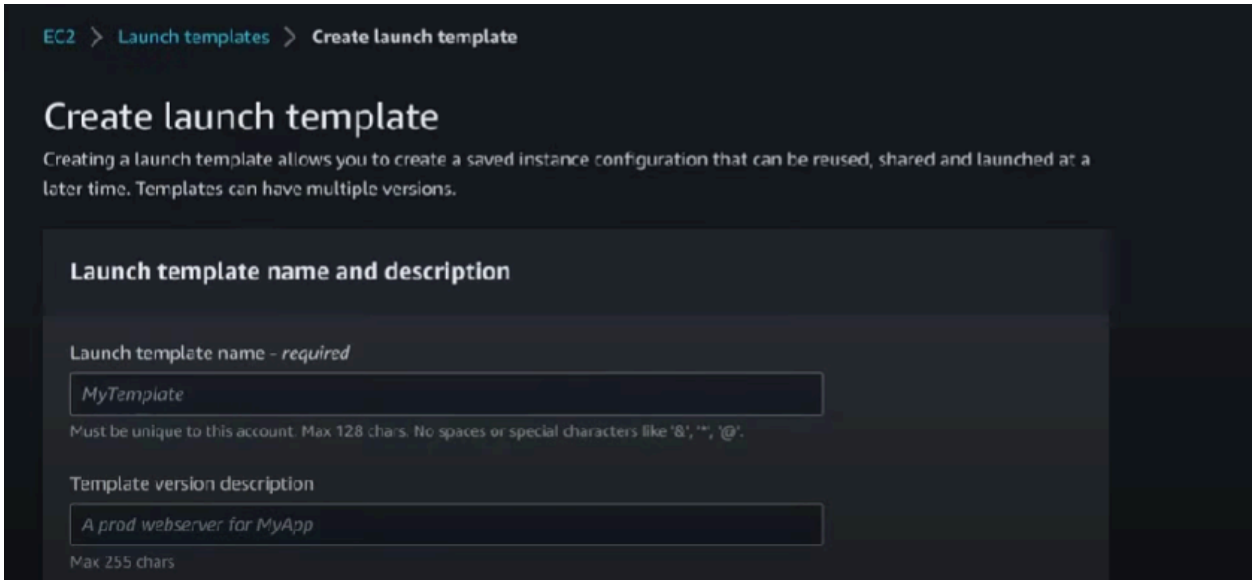
**Thereafter, we need to create Auto scaling group :**

- To create an auto scaling group we need a launch template or launch configuration.
- Select the auto scaling group on the left navbar in the EC2 page and Click on “create auto scaling group” .

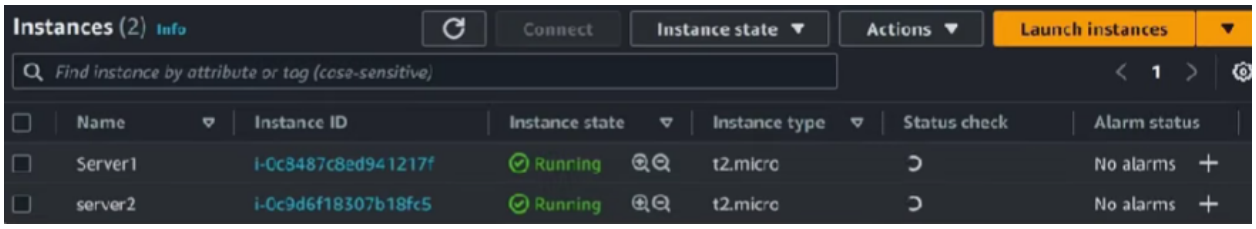


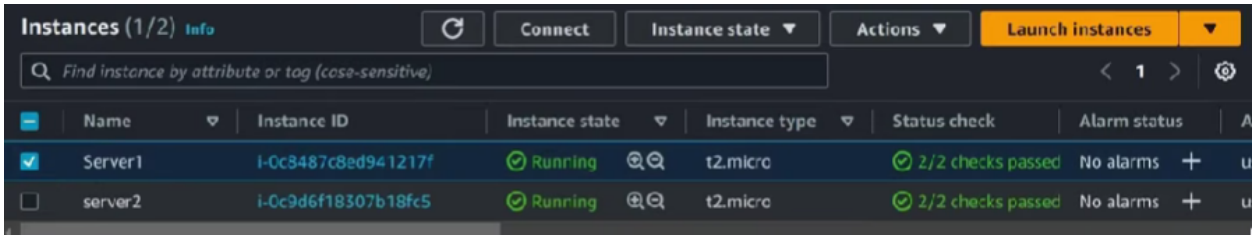




- We will see launch template and configure auto scaling group name and click on “create a launch template”
- Then we will be redirected to a create launch template page

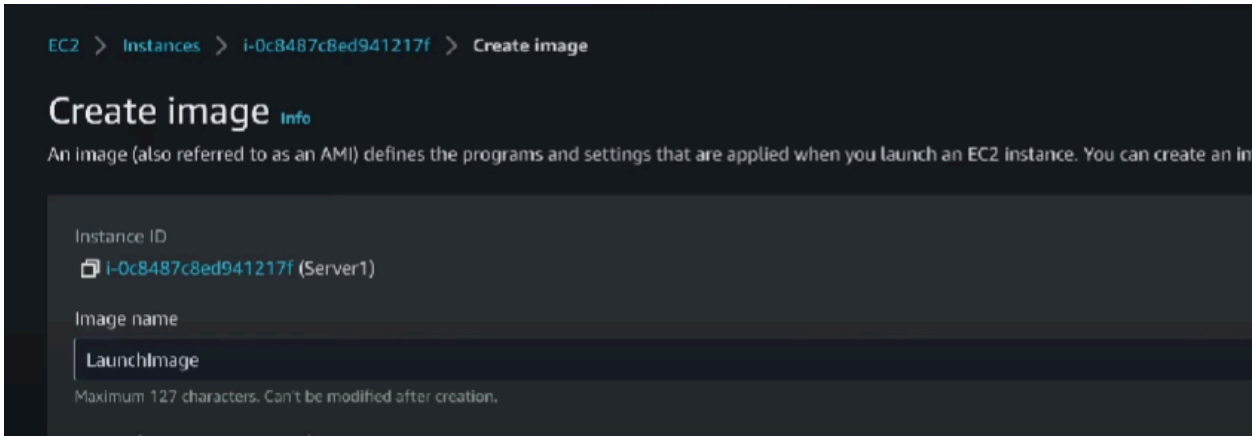


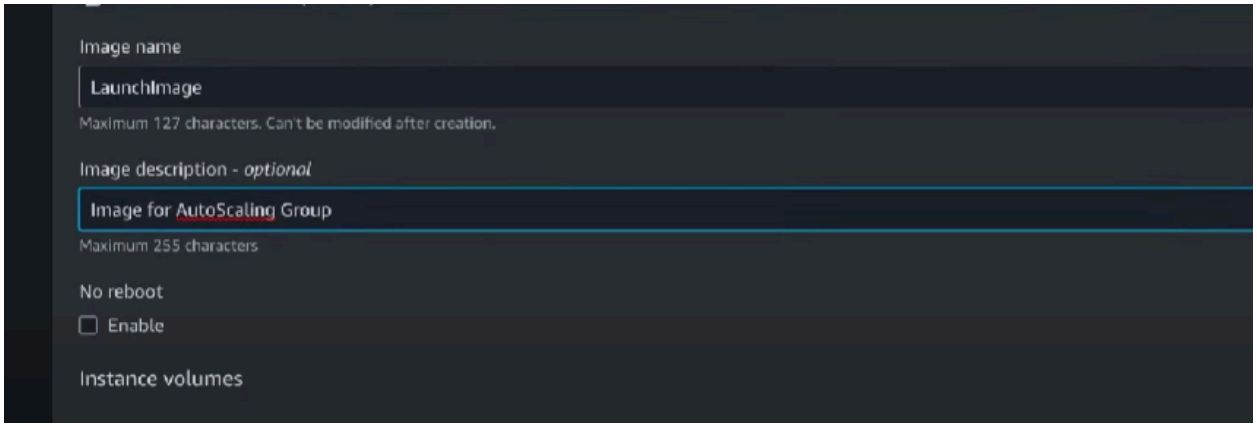

- To create a launch template we need the image of the server so let's go to EC2 and select a server on the checkbox and on the actions button on the top right button select image and template from dropdown and from there select create template.



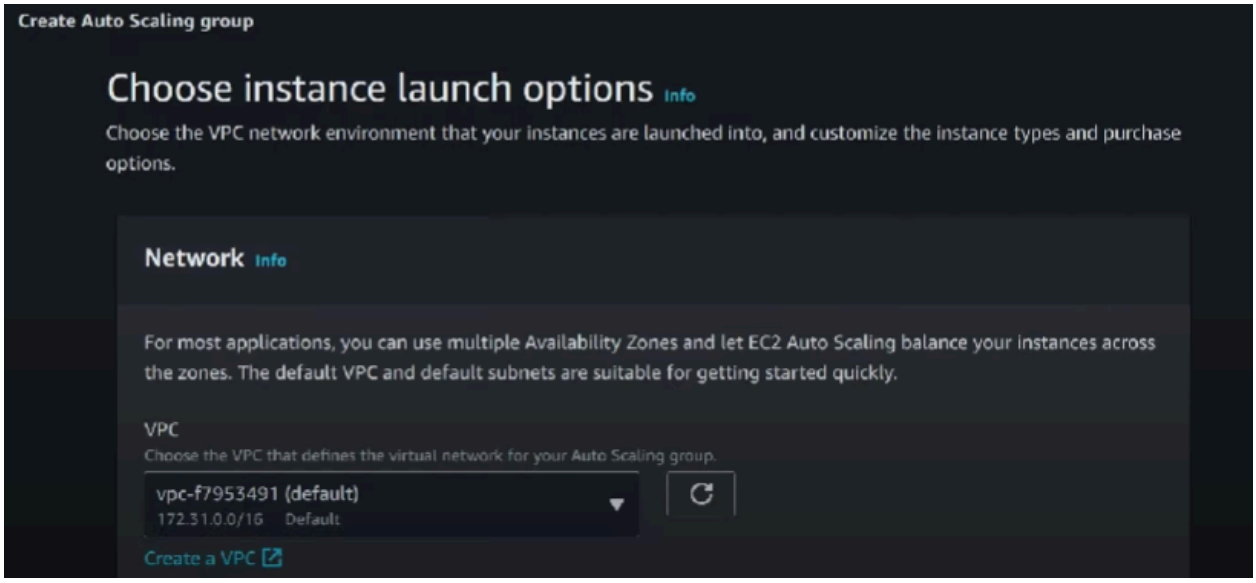




- And then click “create image”

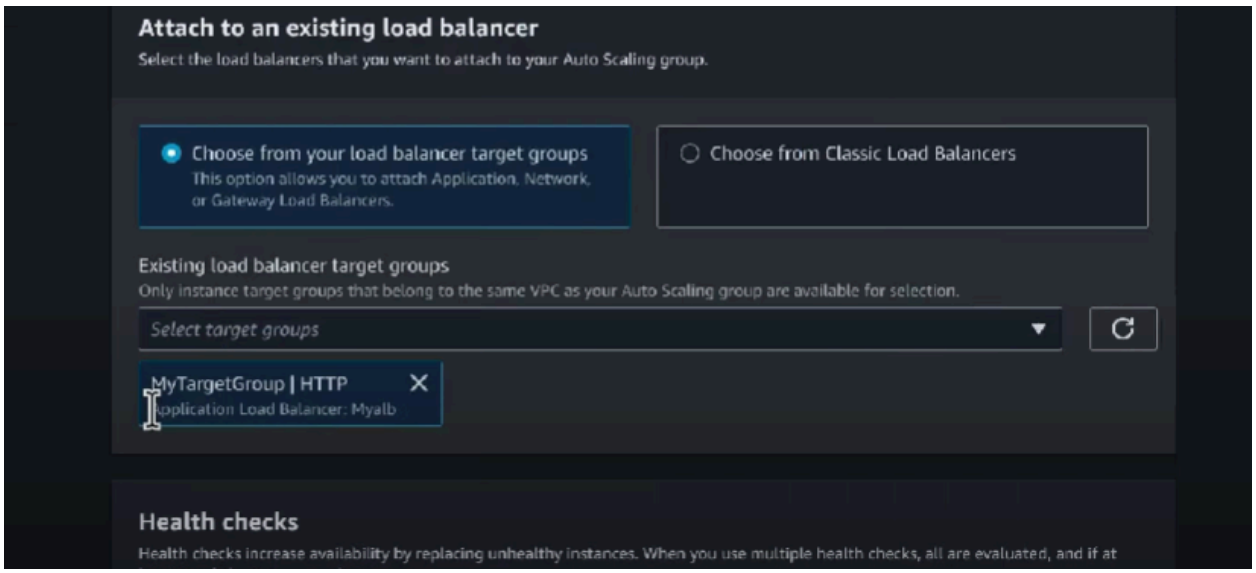



- Now let's create a launch template.
- Give the name and select the ami that we created just now
- Now select the instance t2.micro
- Select security group and click “launch template”

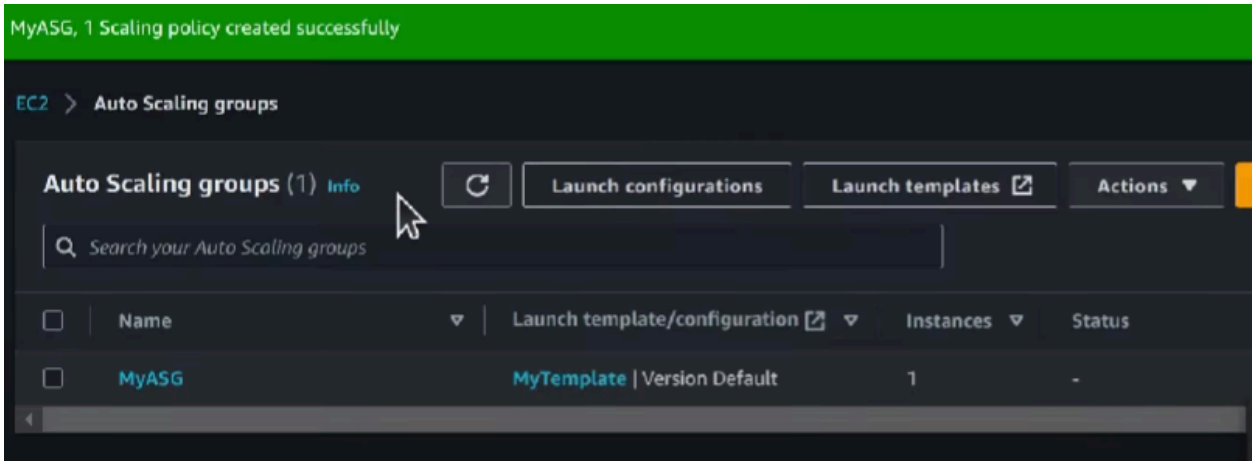



- Now go black to auto scaling group creation and select this launch template . click “next”
- And select the VPC you want to launch, this VPC should be the same where we created the servers, load balancers
- Select all subnets that was created with the VPC and click “next”
- Attach the existing load balancers
- Now we need to enable elastic load balancing health checks and click “next”




- Now we select how many instances auto scaling group to create

- We can select “TARGET TRACKING SCALING POLICY” to launch another server when the previous server reaches a certain percent i.e. if a server reaches 60% it will launch a new server if we set the value 60. We can set the value as our wish depending upon the requirement.
- Click “next” and we will be redirected to the “ add notification” page  and other options which we don't need for now, so we simply skip them and click “ next “ and click on the “create auto scaling group” button to create the auto scaling group.
- Now our auto scaling group is created.




- Inside the ec2 inside auto scaling groups our auto scaling group is created
- We can check now we can see new instances that has been created is also in the target group. This instance is created by an auto scaling group .
- Even if we delete this instance, the auto scaling group will create a new instance again as per the requirements of our project.