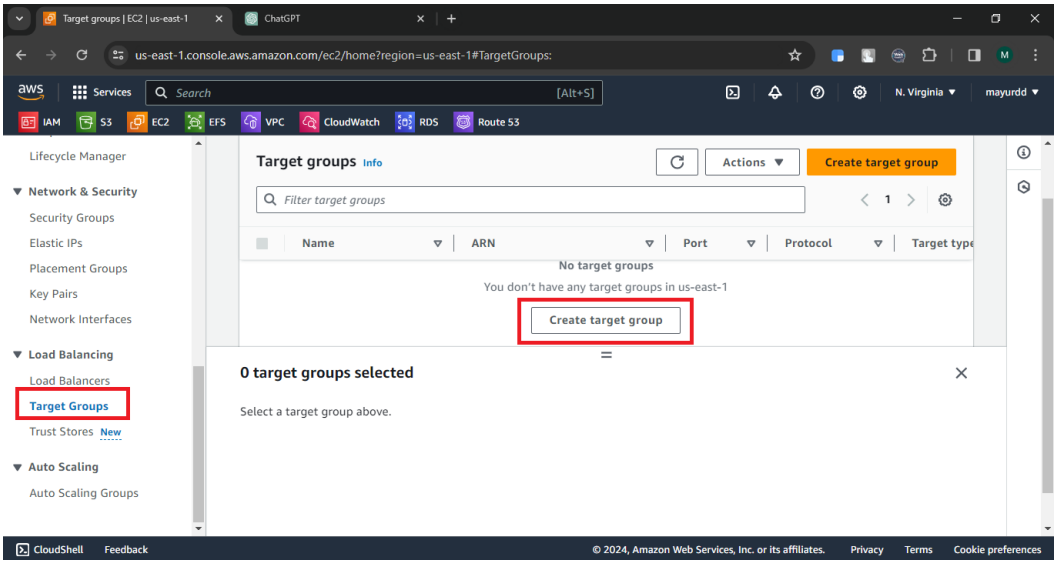
Auto Scaling

What is auto scaling??

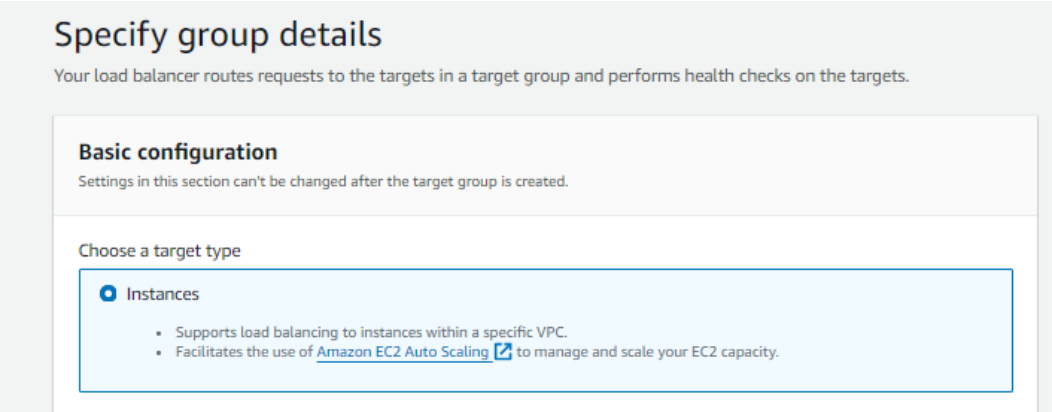
Auto scaling is a process that automatically adjusts the resources (such as servers) allocated to an application based on its current workload. In simpler terms, when demand increases, auto scaling adds more resources to handle the load, and when demand decreases, it reduces resources to save costs.

1. Create an **empty target group**

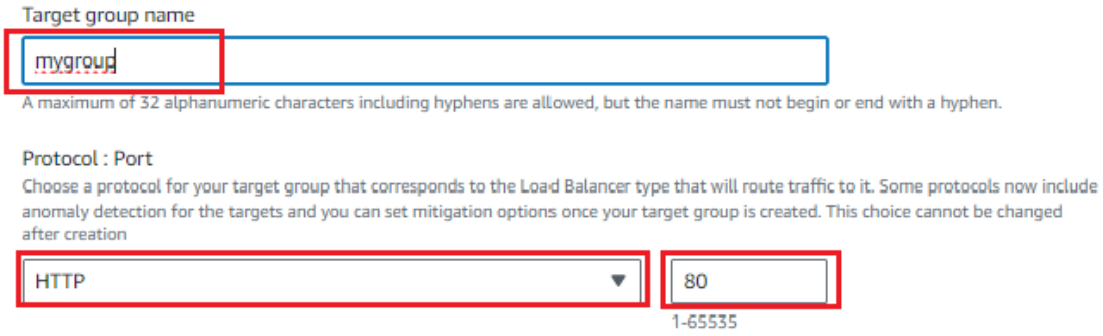
1.1. Click on Create target group



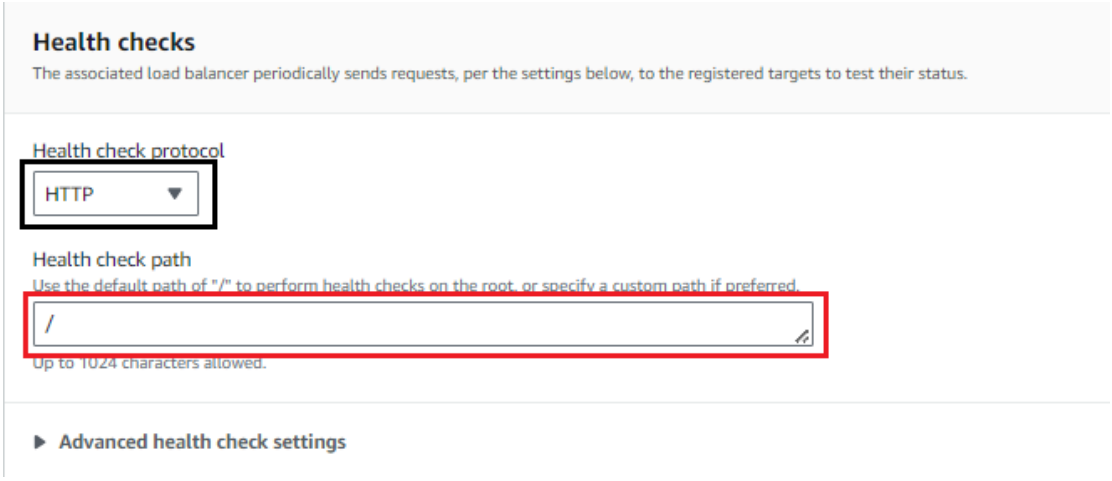
1.2. Select target type (**instances**)



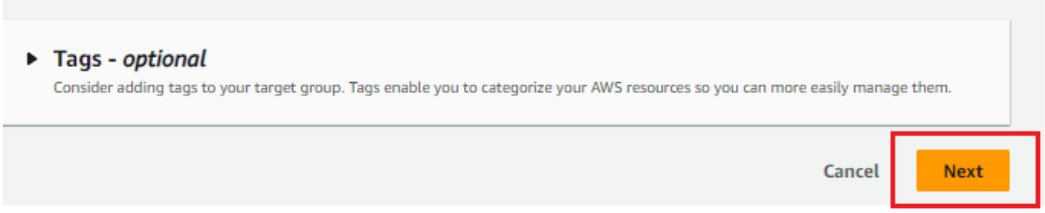
1.3. Give **name, protocol and port** as per your choice



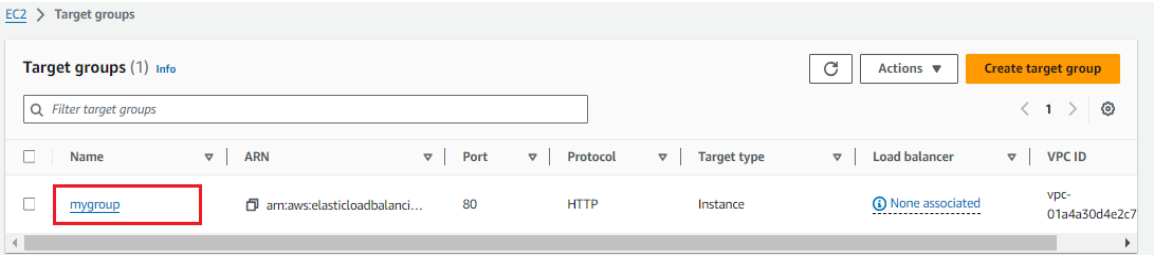
1.4. Select Health checks options



1.5. Click on Next

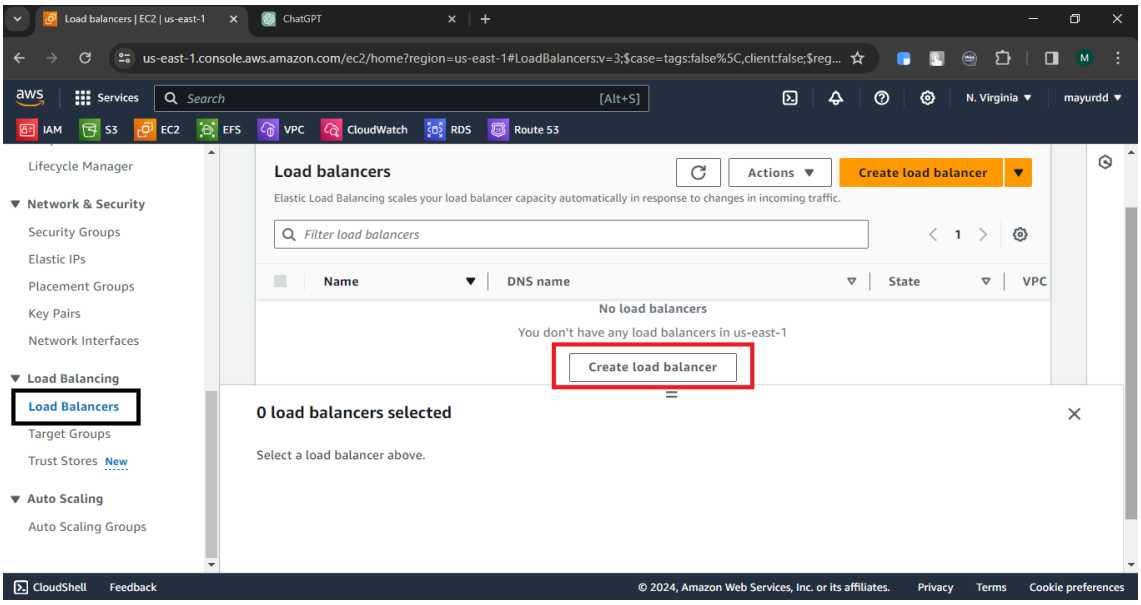


1.6. **Empty Target group Created successfully……**

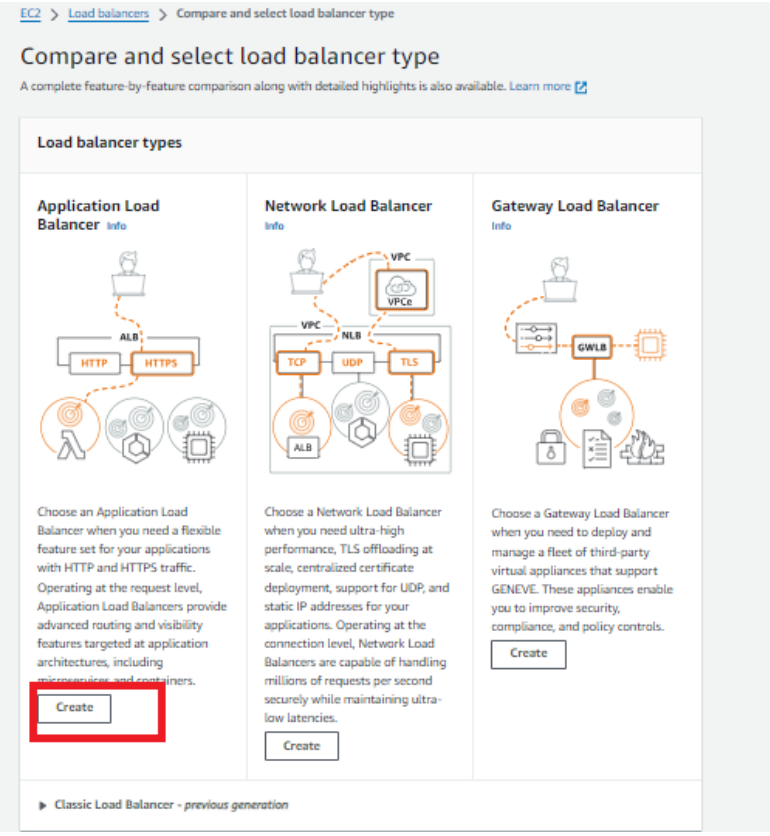


2. Attach Load balancer to Created target group.

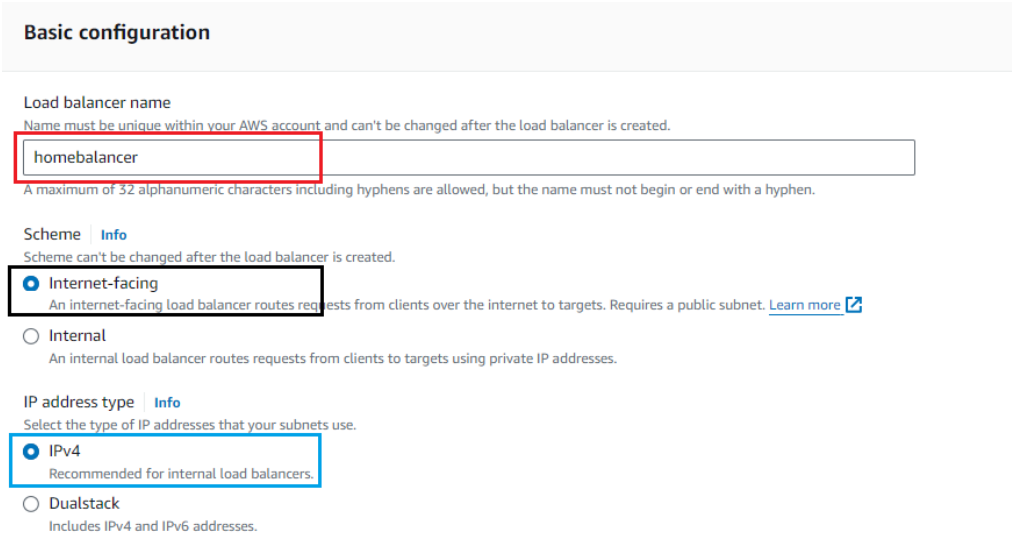
2.1. Click on Create load balancer



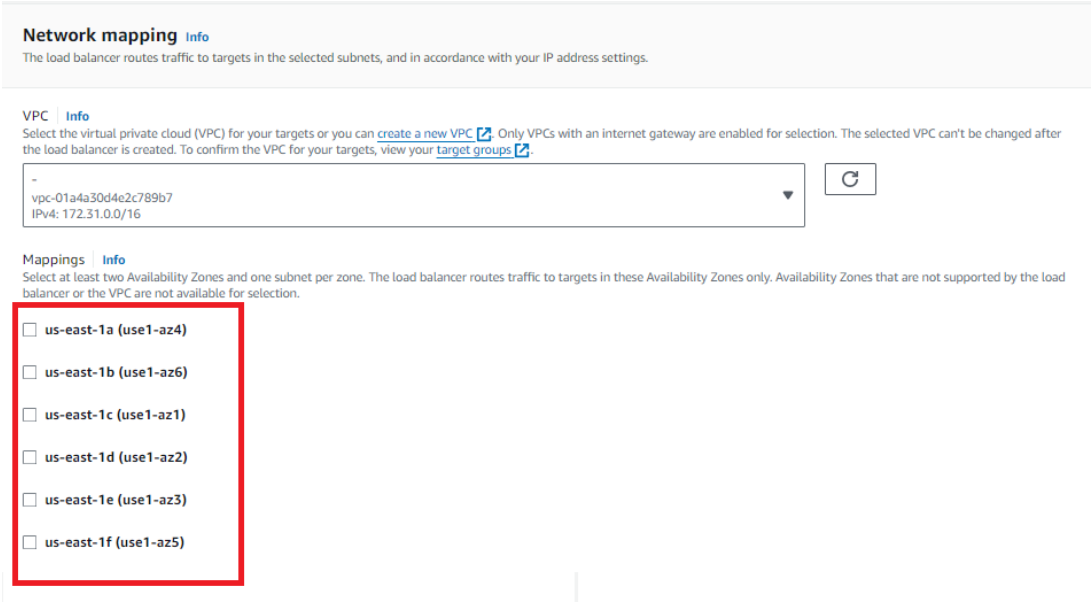
2.2. Click on application load balancer



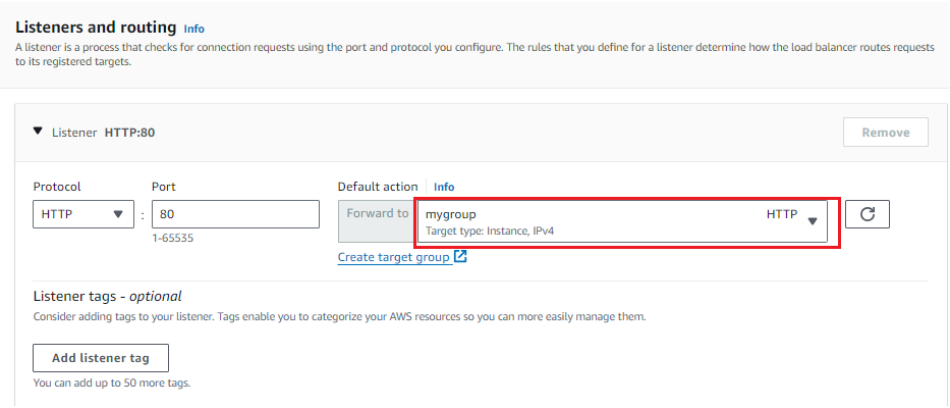
2.3. Basic configuration…..



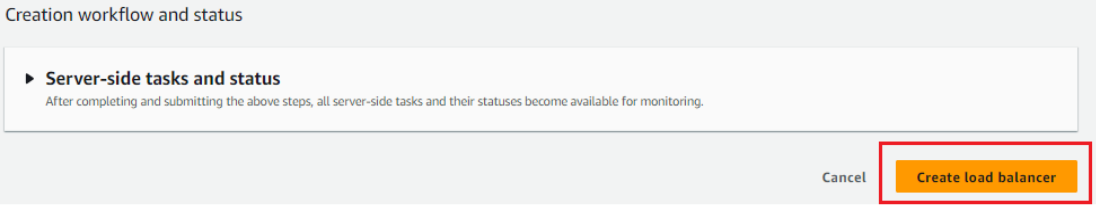
2.4. Click on all zones….



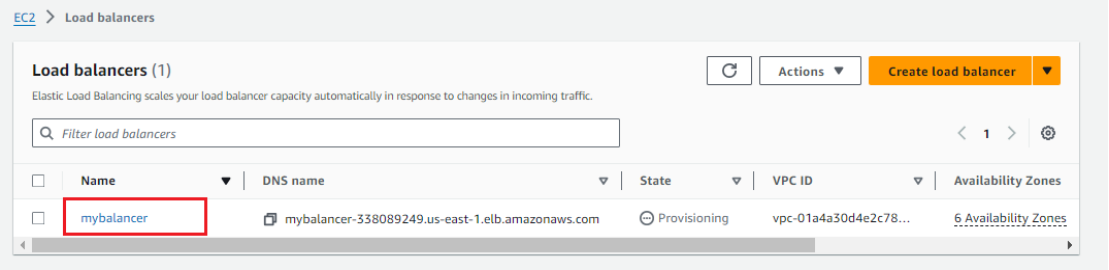
2.5. Select the Target group



2.6. Click on **Create load balancer**

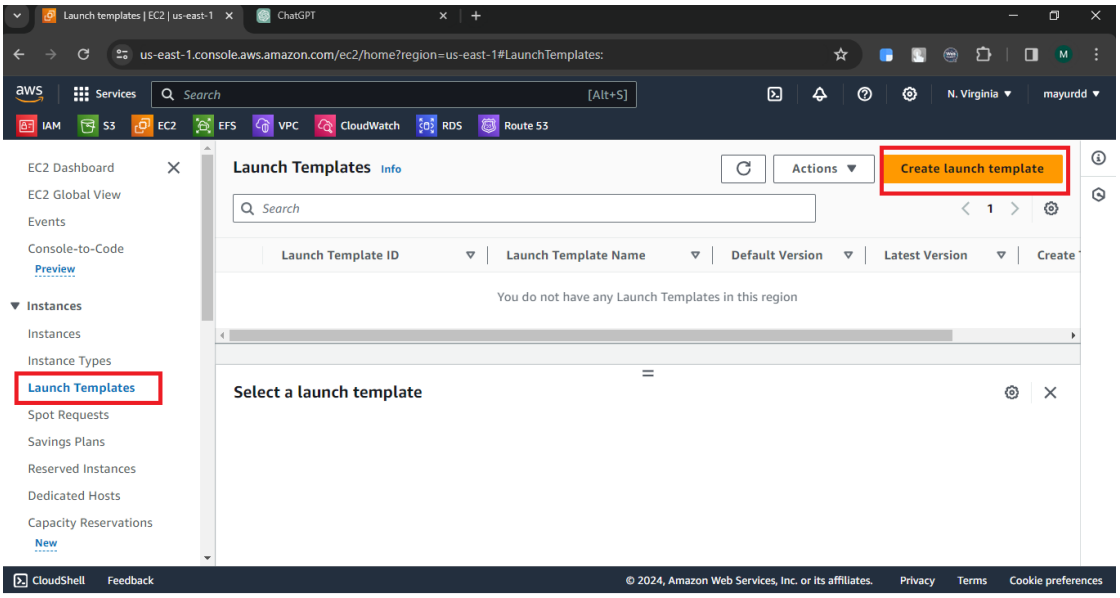


2.7. **Load Balancer Created successfully…..**



3. Create A Template

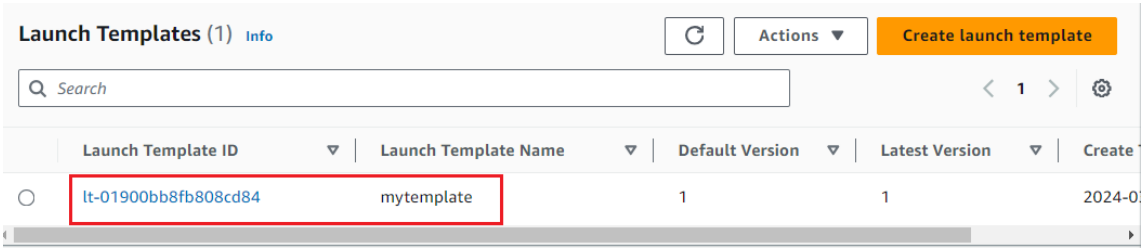
3.1. Click on Create Launch Template



3.2. Specify the required fields, such as…

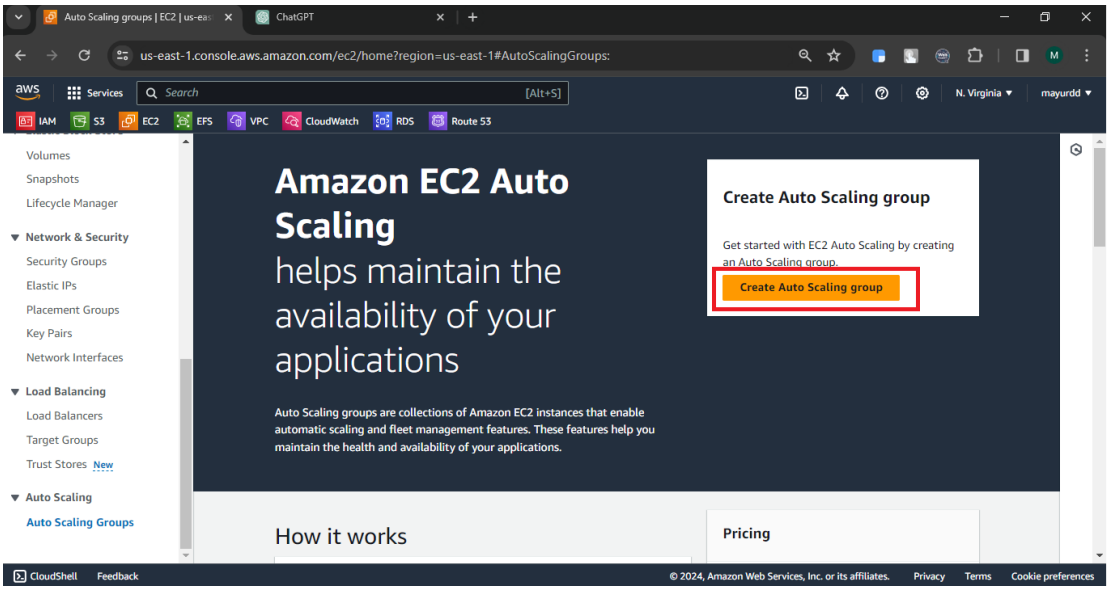
- Template name
- Template description
- AMI
- Instance Type
- Key
- Security group
- Storage
- User-data

3.3. **Template Created successfully….**

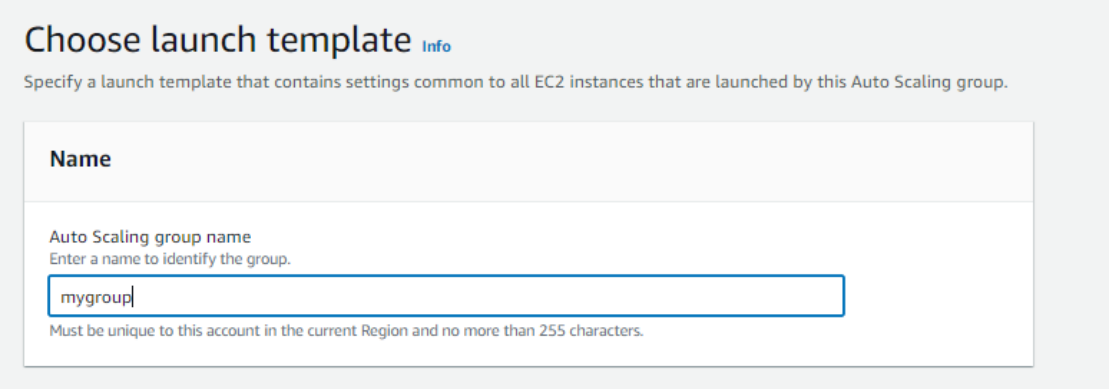


4. Create Auto Scaling group

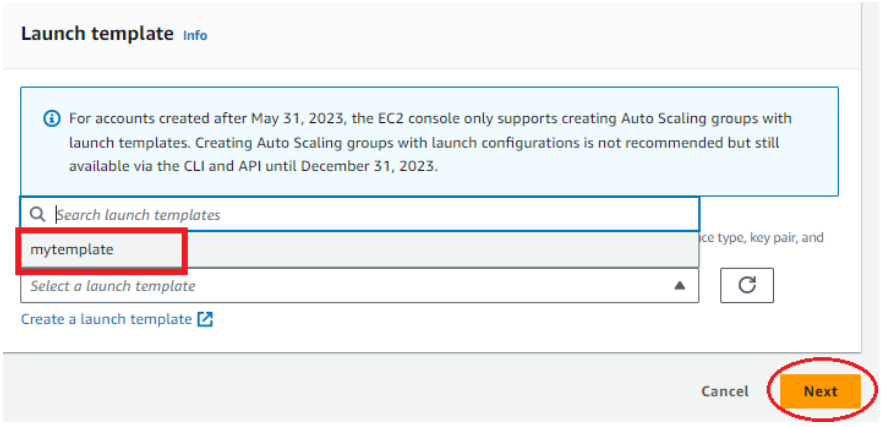
4.1. click on Auto Scaling group option



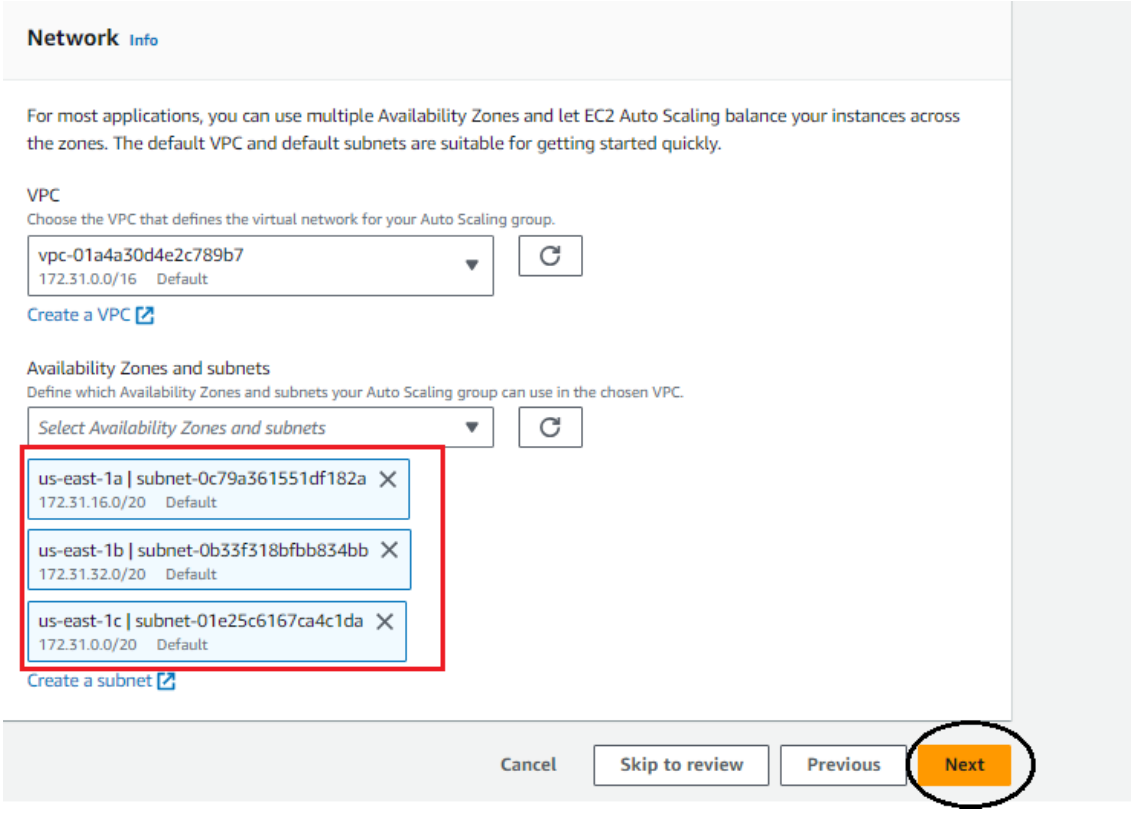
4.2. Give name as per your choice



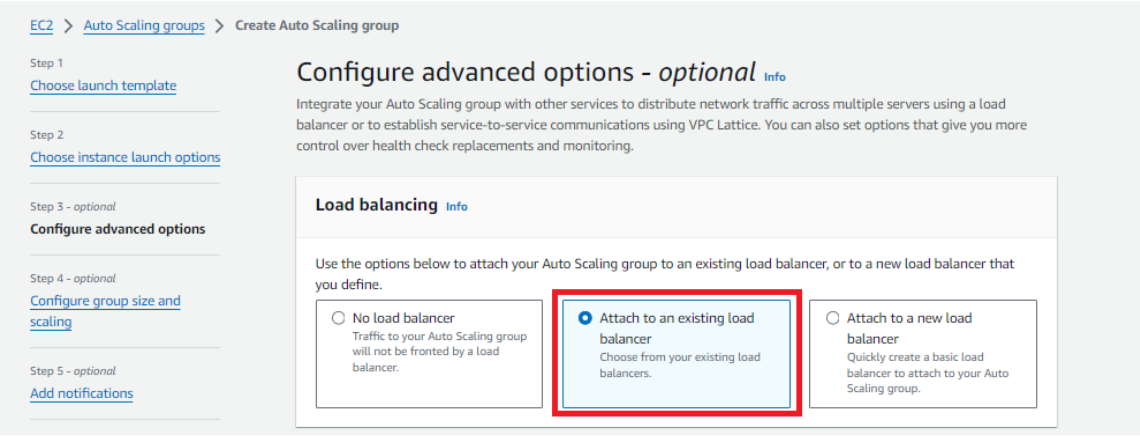
4.3. Select the Template



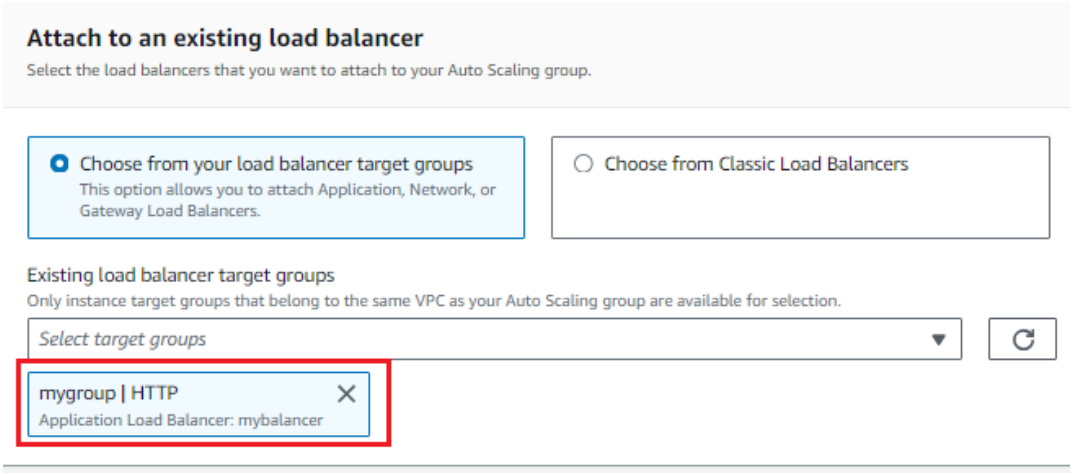
4.4. Select the Availability zones
(Note: - the instance are created automatically in selected zones)



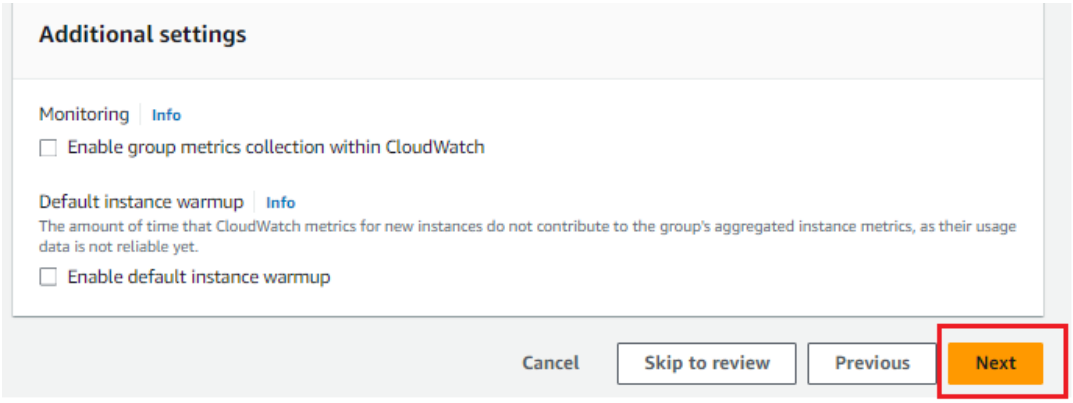
4.5. Click on **<u>Attach to an Existing load balancer</u>**



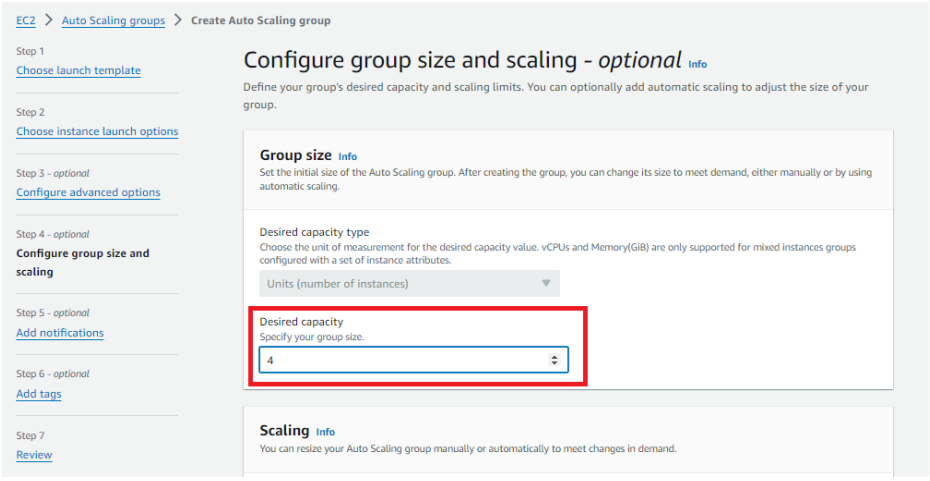
4.6. Select The Target group



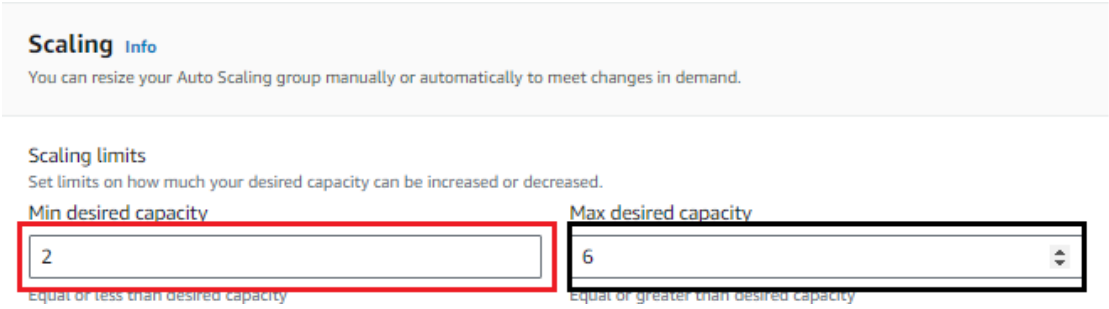
4.7. Click on next



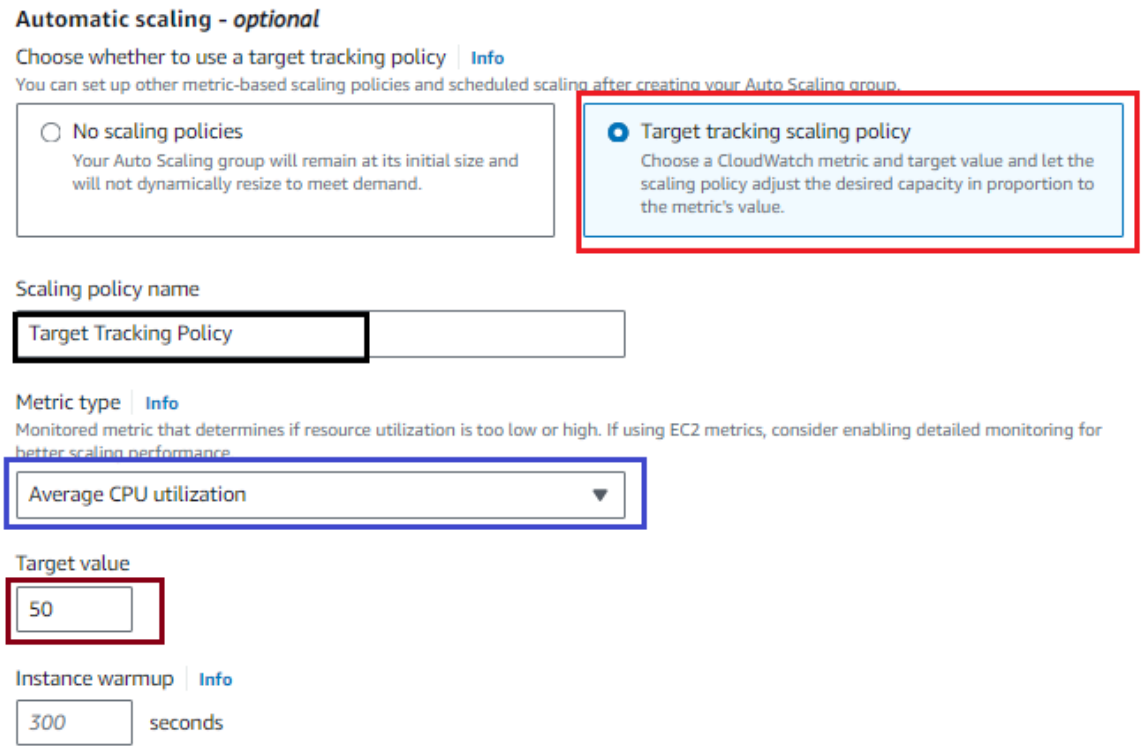
4.8. Select desired capacity =4 **(default instance value)**



4.9. Select min desired capacity= **2** & maximum desired capacity= **6**

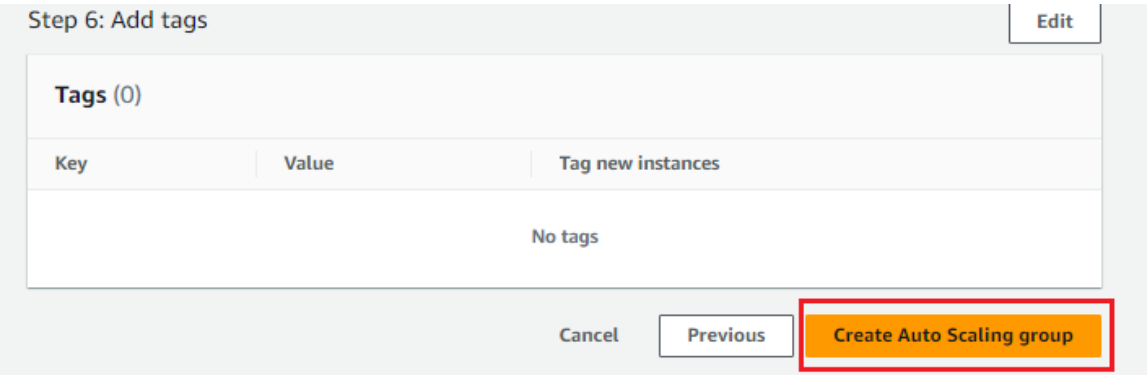


4.10. Click on **<u>Target tracking scaling policy</u>** option

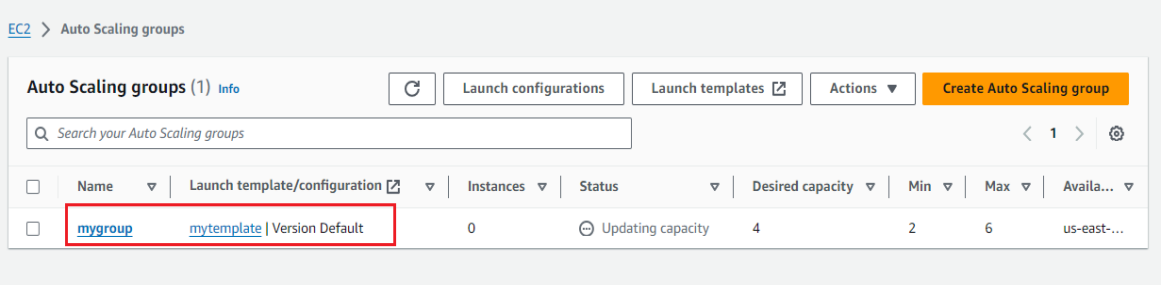


Note: - we are using **cup utilization** for automatic scaling. Means our cup average load increases above 50 % then he automatically launch new instances……

4.11. Click on next, next, and **<u>Create Auto Scaling group</u>** option

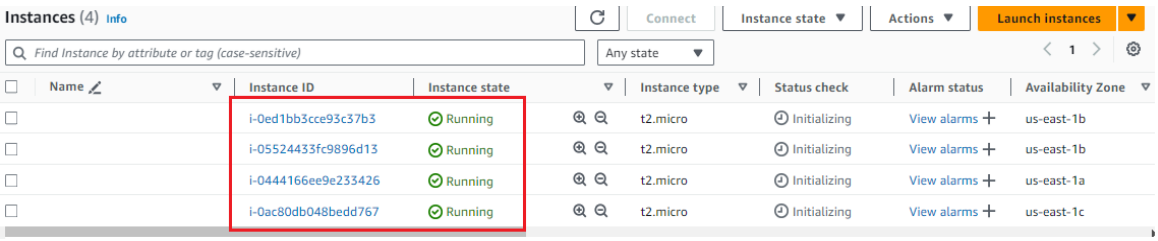


4.12. **Auto scaling group successfully created….**



Note:- now he will create 4 instance automatically with the help of templates….

4.13. Four instance created automatically successfully…..



5. Scaling up using stress command….

1. Increasing the load on instances…

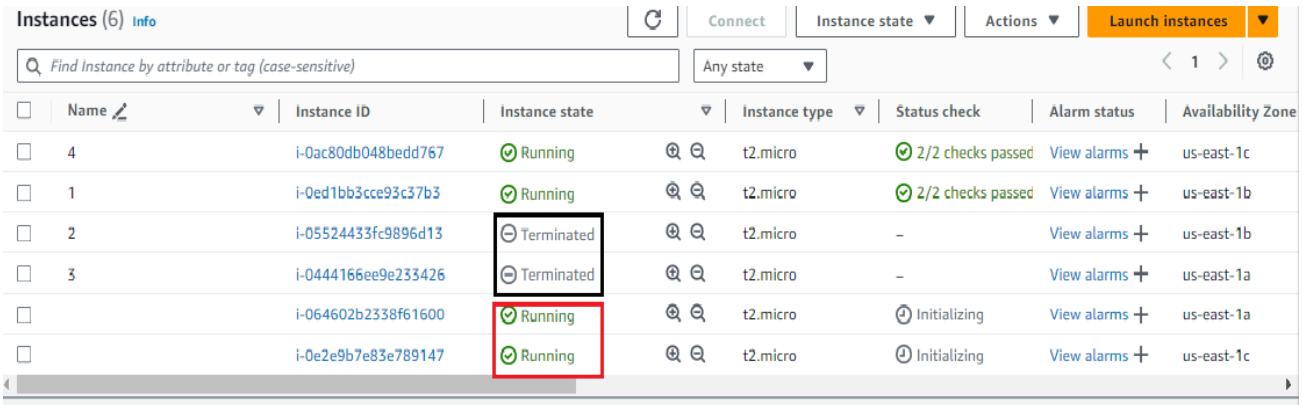
- ssh -i <private_key_name> ec2-user@<ip>
- sudo yum install stress -y
- stress --cpu 88 --io 4 --vm 2 --vm-bytes 128M --timeout 10m &
- clt + r
- **top** (command used for showing live load)

```
top - 08:06:53 up 18 min,  2 users,  load average: 91.20, 47.54, 19.11
Tasks: 203 total,  95 running, 108 sleeping,   0 stopped,   0 zombie
%Cpu(s): 95.3 us,  4.7 sy,  0.0 ni,  0.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :    949.6 total,    310.6 free,    328.3 used,    310.6 buff/cache
MiB Swap:      0.0 total,      0.0 free,      0.0 used.    476.6 avail Mem

   PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
 25726 root      20   0    3512    108      0 R   1.3   0.0   0:02.10 stress
 25733 root      20   0    3512    108      0 R   1.3   0.0   0:02.23 stress
 25734 root      20   0    3512    108      0 R   1.3   0.0   0:02.23 stress
 25735 root      20   0    3512    108      0 R   1.3   0.0   0:02.23 stress
 25737 root      20   0    3512    108      0 R   1.3   0.0   0:02.23 stress
 25738 root      20   0    3512    108      0 R   1.3   0.0   0:02.23 stress
 25740 root      20   0    3512    108      0 R   1.3   0.0   0:02.23 stress
 25743 root      20   0    3512    108      0 R   1.3   0.0   0:02.23 stress
 25745 root      20   0    3512    108      0 R   1.3   0.0   0:02.23 stress
 25749 root      20   0    3512    108      0 R   1.3   0.0   0:02.23 stress
 25752 root      20   0    3512    108      0 R   1.3   0.0   0:02.23 stress
 25758 root      20   0    3512    108      0 R   1.3   0.0   0:02.23 stress
 25760 root      20   0    3512    108      0 R   1.3   0.0   0:02.23 stress
 25761 root      20   0    3512    108      0 R   1.3   0.0   0:02.23 stress
 25764 root      20   0    3512    108      0 R   1.3   0.0   0:02.23 stress
 25765 root      20   0    3512    108      0 R   1.3   0.0   0:02.23 stress
 25766 root      20   0    3512    108      0 R   1.3   0.0   0:02.23 stress
 25771 root      20   0    3512    108      0 R   1.3   0.0   0:02.23 stress
 25772 root      20   0    3512    108      0 R   1.3   0.0   0:02.23 stress
 25773 root      20   0    3512    108      0 R   1.3   0.0   0:02.23 stress
 25774 root      20   0    3512    108      0 R   1.3   0.0   0:02.23 stress
 25775 root      20   0    3512    108      0 R   1.3   0.0   0:02.23 stress
 25776 root      20   0    3512    108      0 R   1.3   0.0   0:02.23 stress
 25781 root      20   0    3512    108      0 R   1.3   0.0   0:02.23 stress
 25783 root      20   0    3512    108      0 R   1.3   0.0   0:02.23 stress
 25785 root      20   0    3512    108      0 R   1.3   0.0   0:02.23 stress
 25787 root      20   0    3512    108      0 R   1.3   0.0   0:02.23 stress
```

1. **Do same process to all instances…..**

2. We can see that after increasing the load, instances are automatically created….



Instances (6) Info

Connect | Instance state | Actions | Launch instances

Find Instance by attribute or tag (case-sensitive) | Any state

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
4	i-0ac80db048bedd767	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1c
1	i-0ed1bb3cce93c37b3	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1b
2	i-05524433fc9896d13	Terminated	t2.micro	–	View alarms +	us-east-1b
3	i-0444166ee9e233426	Terminated	t2.micro	–	View alarms +	us-east-1a
	i-064602b2338f61600	Running	t2.micro	Initializing	View alarms +	us-east-1a
	i-0e2e9b7e83e789147	Running	t2.micro	Initializing	View alarms +	us-east-1c

Note:- when we doesn't assign any load to the instances, they are automatically Terminated…… (**Scaling down process**)
When we as assign extra load to the instances using stress command, the instances are automatically increases….. (**Scaling up process**)