

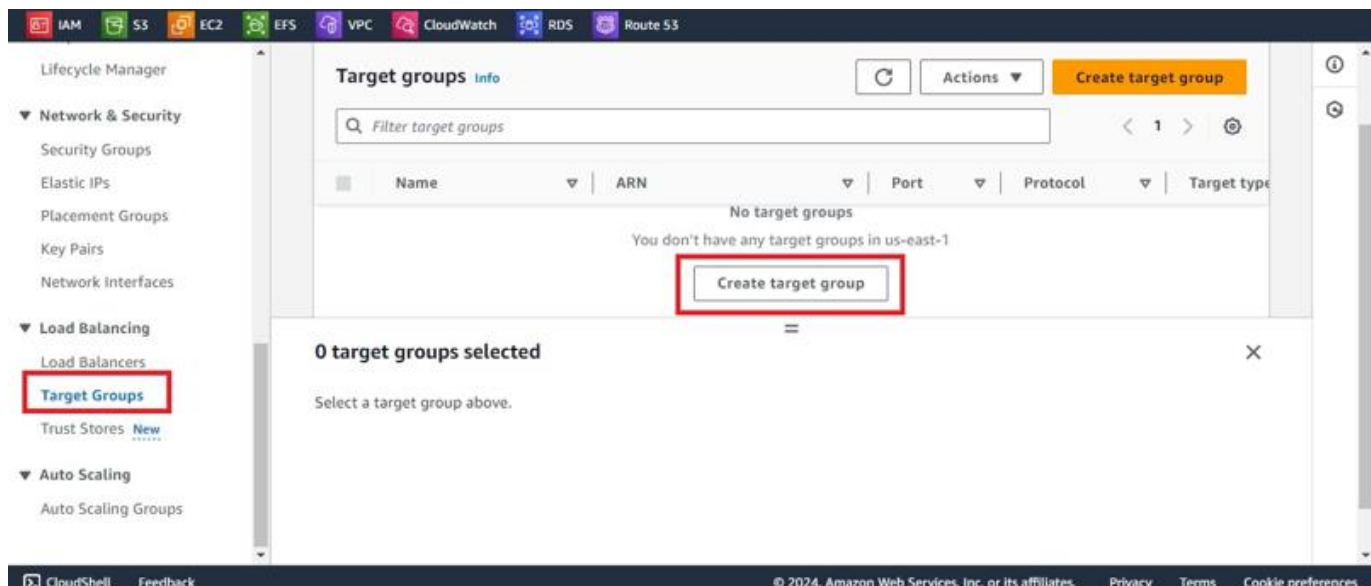
Auto Scaling

What is auto scaling??

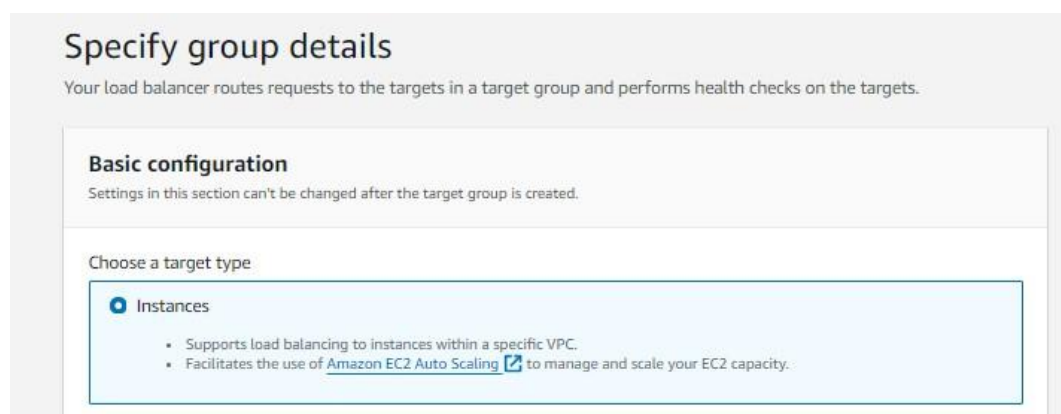
Auto scaling is a process that automatically adjusts the resources (such as servers) allocated to an application based on its current workload. In simpler terms, when demand increases, auto scaling adds more resources to handle the load, and when demand decreases, it reduces resources to save costs.

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

Target group name

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Protocol : Port

Choose a protocol for your target group that corresponds to the Load Balancer type that will route traffic to it. Some protocols now include anomaly detection for the targets and you can set mitigation options once your target group is created. This choice cannot be changed after creation

1-65535

1.4. Select Health checks options

Health checks

The associated load balancer periodically sends requests, per the settings below, to the registered targets to test their status.

Health check protocol

Health check path

Use the default path of "/" to perform health checks on the root, or specify a custom path if preferred.

Up to 1024 characters allowed.

► Advanced health check settings

1.5. Click on Next

► **Tags - optional**

Consider adding tags to your target group. Tags enable you to categorize your AWS resources so you can more easily manage them.

Cancel

1.6. Empty Target group Created successfully.....

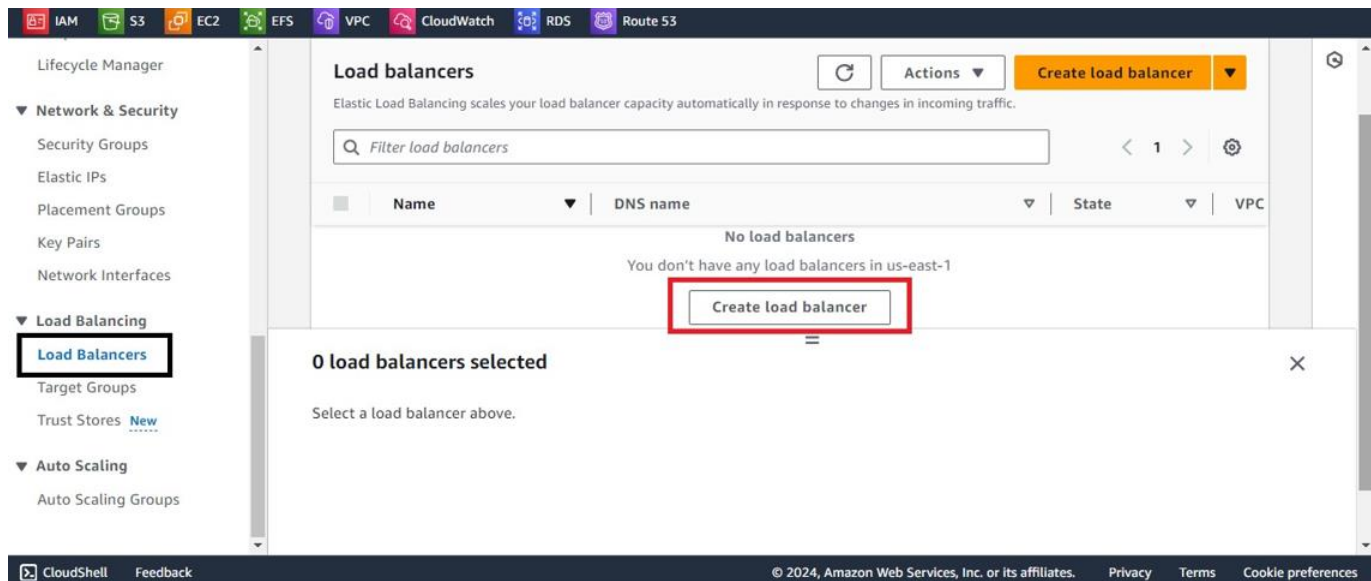
EC2 > Target groups

Target groups (1) Info

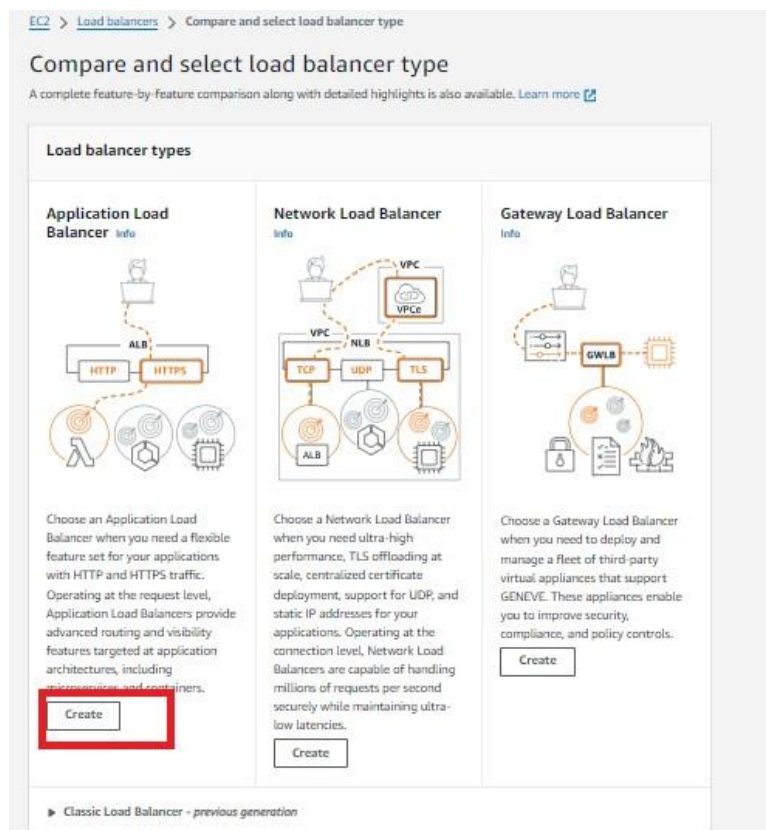
<input type="checkbox"/>	Name	ARN	Port	Protocol	Target type	Load balancer	VPC ID
<input type="checkbox"/>	mygroup	arn:aws:elasticloadbalanci...	80	HTTP	Instance	None associated	vpc-01a4a30d4e2c7

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.....

Basic configuration

Load balancer name

Name must be unique within your AWS account and can't be changed after the load balancer is created.

homebalancer

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

Includes IPv4 and IPv6 addresses.

2.4. Click on all zones.....

Network mapping Info

The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

VPC Info

Select the virtual private cloud (VPC) for your targets or you can [create a new VPC](#). Only VPCs with an internet gateway are enabled for selection. The selected VPC can't be changed after the load balancer is created. To confirm the VPC for your targets, view your [target groups](#).

vpc-01a4a30d4e2c789b7

IPv4: 172.31.0.0/16

Mappings Info

Select at least two Availability Zones and one subnet per zone. The load balancer routes traffic to targets in these Availability Zones only. Availability Zones that are not supported by the load balancer or the VPC are not available for selection.

☐ us-east-1a (use1-az4)

☐ us-east-1b (use1-az6)

☐ us-east-1c (use1-az1)

☐ us-east-1d (use1-az2)

☐ us-east-1e (use1-az3)

☐ us-east-1f (use1-az5)

2.5. Select the Target group

Listeners and routing [info](#)

A listener is a process that checks for connection requests using the port and protocol you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets.

▼ Listener HTTP:80 Remove

Protocol: HTTP Port: 80
1-65535

Default action: [Info](#)

Forward to: mygroup Target type: Instance, IPv4 HTTP ⌂

[Create target group](#)

Listener tags - optional

Consider adding tags to your listener. Tags enable you to categorize your AWS resources so you can more easily manage them.

Add listener tag

You can add up to 50 more tags.

2.6. Click on **Create load balancer**

Creation workflow and status

► **Server-side tasks and status**

After completing and submitting the above steps, all server-side tasks and their statuses become available for monitoring.

Cancel Create load balancer

2.7. Load Balancer Created successfully.....

EC2 > Load balancers

Load balancers (1) ⌂ Actions Create load balancer

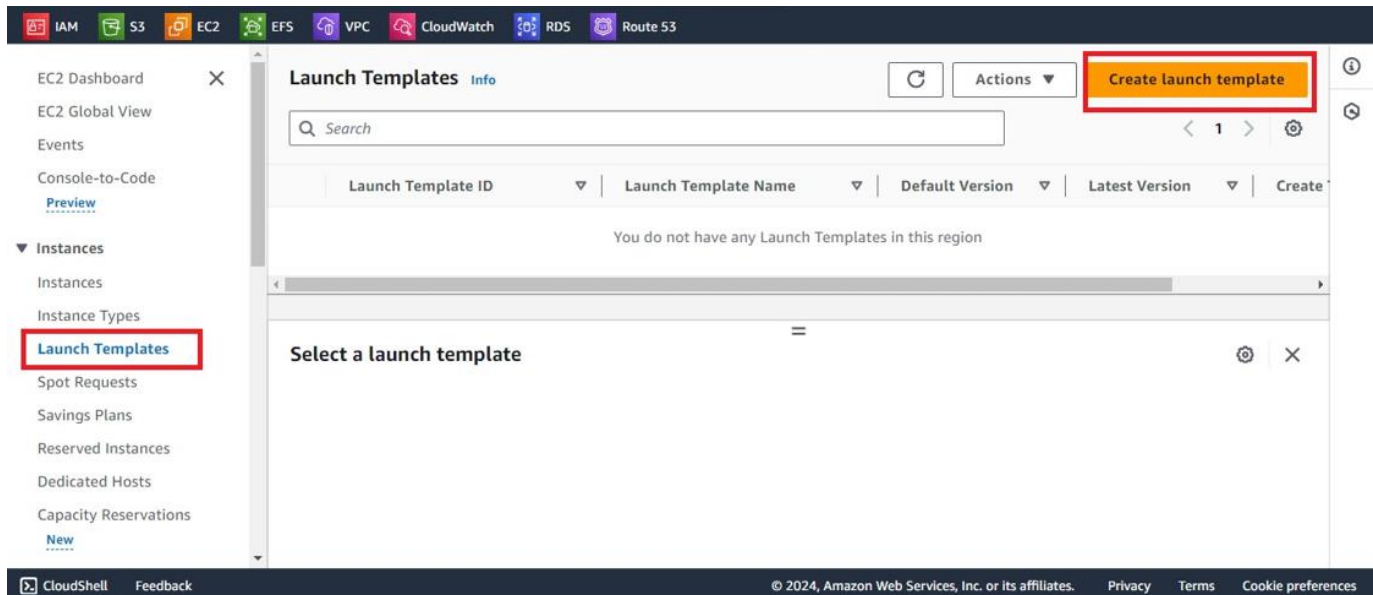
Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

< 1 > ⚙

<input type="checkbox"/>	Name	DNS name	State	VPC ID	Availability Zones
<input type="checkbox"/>	mybalancer	mybalancer-338089249.us-east-1.elb.amazonaws.com	Provisioning	vpc-01a4a30d4e2c78...	6 Availability Zones

3. Create A Template

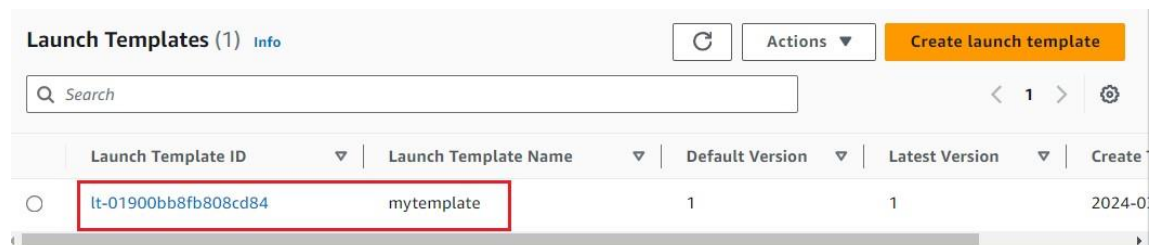
3.1. Click on Create Launch Template



3.2. Specify the required fields, such as...

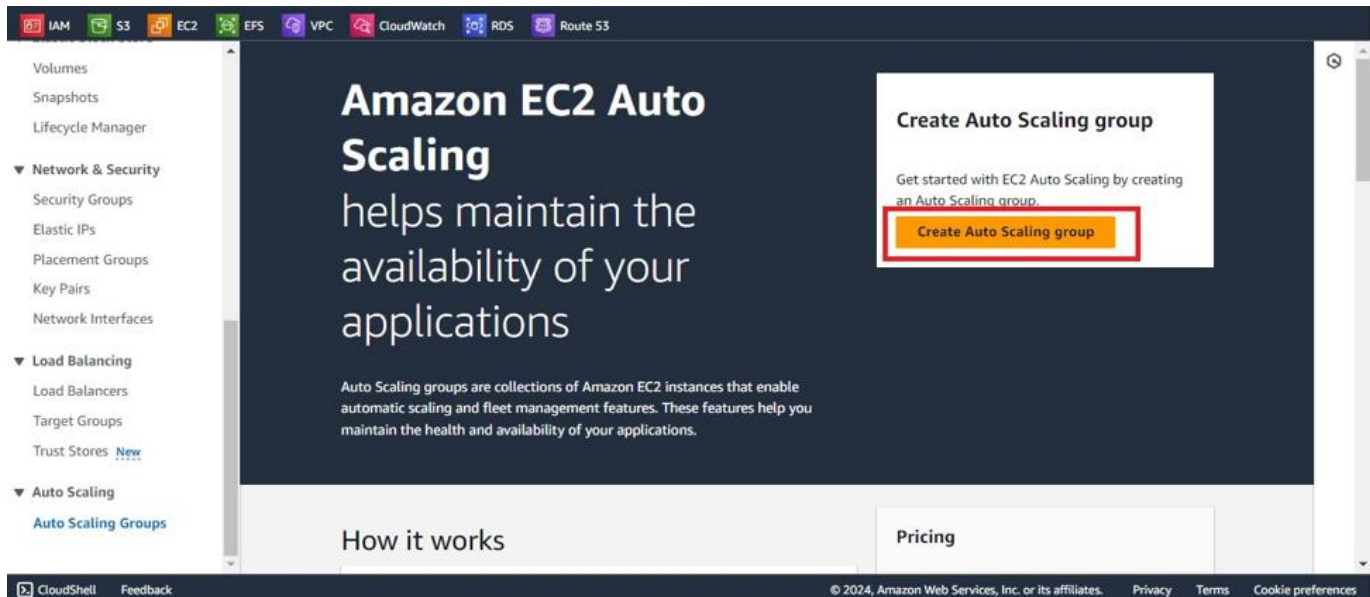
- Template name
- Template description
- AMI
- Instance Type
- Key
- Security group
- Storage
- User-data

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

Choose launch template [Info](#)

Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group.

Name

Auto Scaling group name
Enter a name to identify the group.

Must be unique to this account in the current Region and no more than 255 characters.

4.3. Select the Template

Launch template [Info](#)

For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023.

Select a launch template

Create a launch template [Create](#)

Cancel **Next**

4.4. Select the Availability zones

(Note: - the instance are created automatically in selected zones)

Network Info

For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones. The default VPC and default subnets are suitable for getting started quickly.

VPC
Choose the VPC that defines the virtual network for your Auto Scaling group.

vpc-01a4a30d4e2c789b7
172.31.0.0/16 Default

[Create a VPC](#)

Availability Zones and subnets
Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.

Select Availability Zones and subnets

us-east-1a | subnet-0c79a361551df182a X
172.31.16.0/20 Default

us-east-1b | subnet-0b33f318bfb834bb X
172.31.32.0/20 Default

us-east-1c | subnet-01e25c6167ca4c1da X
172.31.0.0/20 Default

[Create a subnet](#)

Cancel Skip to review Previous **Next**

4.5. Click on Attach to an Existing load balancer

EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1
[Choose launch template](#)

Step 2
[Choose instance launch options](#)

Step 3 - optional
Configure advanced options

Step 4 - optional
[Configure group size and scaling](#)

Step 5 - optional
[Add notifications](#)

Configure advanced options - optional Info

Integrate your Auto Scaling group with other services to distribute network traffic across multiple servers using a load balancer or to establish service-to-service communications using VPC Lattice. You can also set options that give you more control over health check replacements and monitoring.

Load balancing Info

Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.

☐ No load balancer
Traffic to your Auto Scaling group will not be fronted by a load balancer.

☒ **Attach to an existing load balancer**
Choose from your existing load balancers.

☐ Attach to a new load balancer
Quickly create a basic load balancer to attach to your Auto Scaling group.

4.6. Select The Target group

Attach to an existing load balancer
Select the load balancers that you want to attach to your Auto Scaling group.

☒ Choose from your load balancer target groups
This option allows you to attach Application, Network, or Gateway Load Balancers.

☐ Choose from Classic Load Balancers

Existing load balancer target groups
Only instance target groups that belong to the same VPC as your Auto Scaling group are available for selection.

Select target groups

mygroup | HTTP
Application Load Balancer: mybalancer

Next

4.7. Click on next

Additional settings

Monitoring [Info](#)

☐ Enable group metrics collection within CloudWatch

Default instance warmup [Info](#)

The amount of time that CloudWatch metrics for new instances do not contribute to the group's aggregated instance metrics, as their usage data is not reliable yet.

☐ Enable default instance warmup

Cancel Skip to review Previous **Next**

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

EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1
[Choose launch template](#)

Step 2
[Choose instance launch options](#)

Step 3 - optional
[Configure advanced options](#)

Step 4 - optional
Configure group size and scaling

Step 5 - optional
[Add notifications](#)

Step 6 - optional
[Add tags](#)

Step 7
[Review](#)

Configure group size and scaling - optional [Info](#)

Define your group's desired capacity and scaling limits. You can optionally add automatic scaling to adjust the size of your group.

Group size [Info](#)

Set the initial size of the Auto Scaling group. After creating the group, you can change its size to meet demand, either manually or by using automatic scaling.

Desired capacity type

Choose the unit of measurement for the desired capacity value. vCPUs and Memory(GiB) are only supported for mixed instances groups configured with a set of instance attributes.

Units (number of instances)

Desired capacity
Specify your group size.

4

Scaling [Info](#)

You can resize your Auto Scaling group manually or automatically to meet changes in demand.

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

Scaling [Info](#)
You can resize your Auto Scaling group manually or automatically to meet changes in demand.

Scaling limits
Set limits on how much your desired capacity can be increased or decreased.

Min desired capacity
2
Equal or less than desired capacity

Max desired capacity
6
Equal or greater than desired capacity

4.10. Click on **Target tracking scaling policy** option

Automatic scaling - optional
Choose whether to use a target tracking policy [Info](#)
You can set up other metric-based scaling policies and scheduled scaling after creating your Auto Scaling group.

☐ No scaling policies
Your Auto Scaling group will remain at its initial size and will not dynamically resize to meet demand.

☒ **Target tracking scaling policy**
Choose a CloudWatch metric and target value and let the scaling policy adjust the desired capacity in proportion to the metric's value.

Scaling policy name
Target Tracking Policy

Metric type [Info](#)
Monitored metric that determines if resource utilization is too low or high. If using EC2 metrics, consider enabling detailed monitoring for better scaling performance.

Average CPU utilization

Target value
50

Instance warmup [Info](#)
300 seconds

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 **Create Auto Scaling group** option

Step 6: Add tags Edit

Tags (0)

Key	Value	Tag new instances
No tags		

Cancel Previous **Create Auto Scaling group**

4.12. Auto scaling group successfully created....

EC2 > Auto Scaling groups

Auto Scaling groups (1) [Info](#) Refresh Launch configurations Launch templates Actions **Create Auto Scaling group**

<input type="checkbox"/>	Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zone
<input type="checkbox"/>	mygroup	mytemplate Version Default	0	Updating capacity	4	2	6	us-east-1b

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

4.13. Four instance created automatically successfully.....

Instances (4) [Info](#) Refresh Connect Instance state Actions **Launch instances**

Any state

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input type="checkbox"/>		i-0ed1bb3cce93c37b3	Running	t2.micro	Initializing	View alarms	us-east-1b
<input type="checkbox"/>		i-05524433fc9896d13	Running	t2.micro	Initializing	View alarms	us-east-1b
<input type="checkbox"/>		i-0444166ee9e233426	Running	t2.micro	Initializing	View alarms	us-east-1a
<input type="checkbox"/>		i-0ac80db048bedd767	Running	t2.micro	Initializing	View alarms	us-east-1c

5. Scaling up using stress command....

1. Increasing the load on instances...

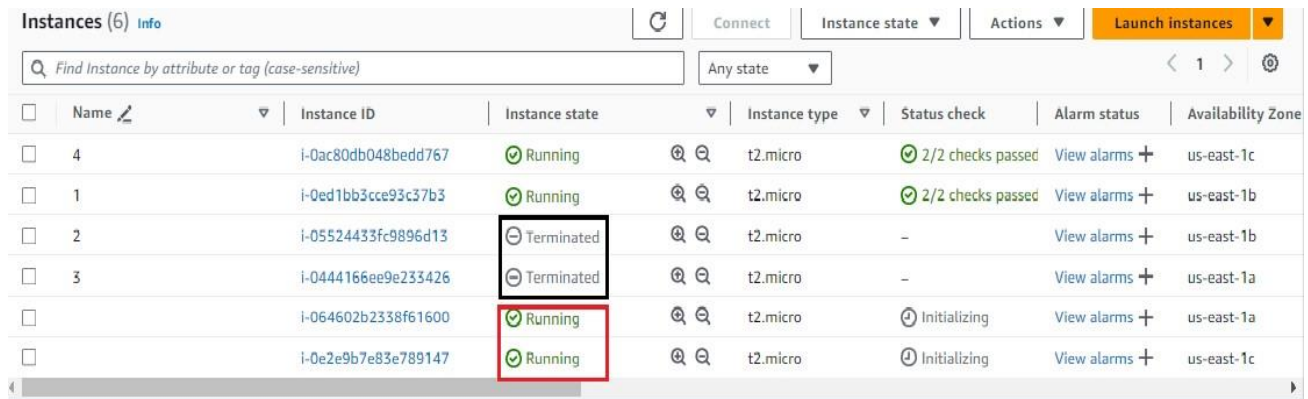
- `ssh -i <private_key_name> ec2-user@<ip>`
- `sudo yum install stress -y`
- `stress --cpu 88 --io 4 --vm 2 --vm-bytes 128M --timeout 10m &`
□ `clt + r`
- **top** (command used for showing live load)

```
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....



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