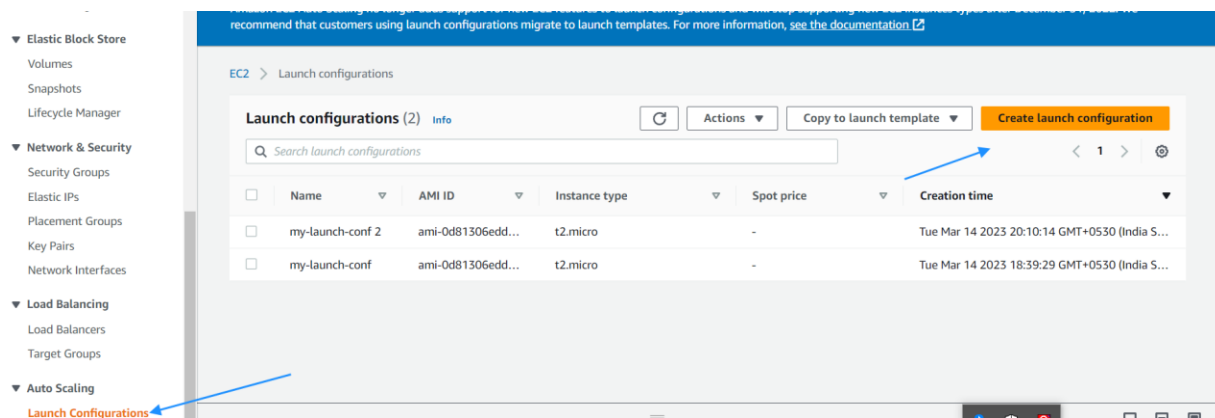


## Auto Scalling with load balancer

### Steps :-

- 1) Install two instances with ssh and http inbound traffic allow using user data as below  
#!/bin/bash  
sudo -i  
yum install httpd -y  
echo "<h1> this is home page \$HOSTNAME</h1>" >  
/var/www/html/index.html  
systemctl start httpd  
systemctl enable httpd
- 2) Now in load balancer create target group using these two instances
- 3) Now create load balancer using these home-tg target group
- 4) Now in auto scaling create launch template same as home instances (ssh and http traffic allow),(same user data like home page)



- 5) Give name to the launch temp → select amazon linux machine image  
→select instance type t2.micro

**Launch configuration name**

Name

apache-server-launch-conf

**Amazon machine image (AMI)** [Info](#)

AMI

amzn2-ami-kernel-5.10-hvm-2.0.20230307.0-x86\_64-gp2

**Instance type** [Info](#)

Instance type

t2.micro (1 vCPUs, 1 GiB, EBS Only)

Choose instance type

## 6) Enable instance detailed monitoring

**Monitoring** [Info](#)

☒ Enable EC2 instance detailed monitoring within CloudWatch

## 7) Select existing security group which has http and ssh inbound traffic allowed

**Security groups** [Info](#)

Assign a security group

☐ Create a new security group

☒ Select an existing security group

**Security groups** [Copy to new](#) [View rules](#)

Search security groups


	Security group ID	Name	VPC ID	Description
<input checked="" type="checkbox"/>	sg-0f75112c17f47bd37	my-security-group	vpc-020ef4b35bc15d210	my-sg
<input type="checkbox"/>	sg-0f0524446c705eadd	launch-wizard-7	vpc-020ef4b35bc15d210	launch-wizard-7 created 2023-03-10T10:12:48.922Z


## 8) Select existing key-pair and create launch configuration


**Key pair (login)** [Info](#)

Key pair options

Choose an existing key pair

Existing key pair  mumbaikey

☒ I acknowledge that I have access to the selected private key file (mumbaikey.pem), and that without this file, I won't be able to log into my instance. 

[Cancel](#) [Create launch configuration](#) 

9) Now go to auto scaling → create an auto scaling group

10) Give auto scaling group name

Step 1  
**Choose launch template or configuration**

Step 2  
Choose instance launch options


Step 3 - optional  
Configure advanced options

Step 4 - optional  
Configure group size and scaling policies

**Choose launch template or configuration** [Info](#)

Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group. If you currently use launch configurations, you might consider migrating to launch templates.


**Name**

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

apache-auto-scaling-with-load-balancer


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


11) Switch to launch configuration

**Launch template** [Info](#) 

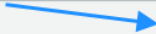
**Launch template**

Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.

Select a launch template 

[Create a launch template](#) 



[Switch to launch configuration](#)

[Cancel](#) [Next](#) 

12) Select existing launch template which has home user data in it and click on next

## Launch configuration [Info](#)

[Switch to launch template](#)

 Instead of using launch configurations to create your EC2 Auto Scaling groups, we recommend that you use launch templates and make use of the Auto Scaling guidance option. For more information on migrating launch configurations and using launch templates, [see the documentation](#) 

### Launch configuration

Choose a launch configuration that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.

apache-server-launch-conf ▼



[Create a launch configuration](#) 

Launch configuration

apache-server-launch-conf

AMI ID

ami-0d81306eddc614a45

Date created

Wed Mar 15 2023 19:12:17  
GMT+0530 (India Standard Time)

Security groups

[sg-0f75112c17f47bd37](#) 

Instance type

t2.micro

Key pair name

mumbaikey

Cancel

Next

## 13) Select default VPC and select all subnet and click on next

### 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-020ef4b35bc15d210 ▼

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 ▼



ap-south-1a | subnet-049192010405dc2e7 ✕

172.31.32.0/20 Default

ap-south-1b | subnet-05c57a12dc04a8b6b ✕

172.31.0.0/20 Default

ap-south-1c | subnet-0347d8e71b090eb23 ✕

172.31.16.0/20 Default

[Create a subnet](#) 

Cancel

Skip to review

Previous

Next

- 14) Now in step 3 select existing load balancer → select existing target group which is home-TG as shown in fig

[Choose instance launch options](#)

Step 3 - optional  
**Configure advanced options**

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

Step 5 - optional  
[Add notifications](#)

Step 6 - optional  
[Add tags](#)

Step 7  
[Review](#)

### Load balancing - optional [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.

### 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

home-tg | HTTP ×  
Application Load Balancer: my-alb

- 15) Now reduce health check grace period to 100 s → in additional setting enable group metrics collection within cloudwatch and click on next

### Health checks - optional

Health check type [Info](#)

EC2 Auto Scaling automatically replaces instances that fail health checks. If you enabled load balancing, you can enable ELB health checks in addition to the EC2 health checks that are always enabled.

☒ EC2 ☐ ELB

Health check grace period

The amount of time until EC2 Auto Scaling performs the first health check on new instances after they are put into service.

seconds

### Additional settings - optional

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](#)

- 16) Now in step 4 give desired capacity 4 min capacity 1 max capacity 6

Step 2  
[Choose instance launch options](#)

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

Step 4 - optional  
**Configure group size and scaling policies**

Step 5 - optional  
[Add notifications](#)

Step 6 - optional  
[Add tags](#)

### Group size - optional [Info](#)

Specify the size of the Auto Scaling group by changing the desired capacity. You can also specify minimum and maximum capacity limits. Your desired capacity must be within the limit range.

Desired capacity

Minimum capacity

Maximum capacity

- 17) Now in scaling policy select target tracking scaling policy → metrics type **cpu utilization** → target value 50 → instance need 100 and click on next

**Scaling policies - optional**

Choose whether to use a scaling policy to dynamically resize your Auto Scaling group to meet changes in demand. [Info](#)

☒ Target tracking scaling policy  
 Choose a desired outcome and leave it to the scaling policy to add and remove capacity as needed to achieve that outcome.

☐ None

Scaling policy name

Metric type

Target value

Instances needed  
 seconds warm up before including in metric

☐ Disable scale in to create only a scale-out policy

**Instance scale-in protection - optional**

Instance scale-in protection  
 If protect from scale in is enabled, newly launched instances will be protected from scale in by default.

☐ Enable instance scale-in protection

Cancel Skip to review Previous **Next**

- 18) Give notification if you want
- 19) Give tags if needed
- 20) Review and create auto scaling group
- 21) Now go to load balance and copy DNS and hit in new tab. It will distribute traffic across different 4 desired instance

**Load balancers (1)**  
 Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

Filter by property or value

<input type="checkbox"/>	Name	DNS name	State	VPC ID	Availability Zones	Type	Date created	Inst
<input type="checkbox"/>	my-alb	my-alb-1098923949.ap-so...	Active	vpc-020ef4b35bc15d210	3 Availability Zones	application	March 14, 2023, 16:45 (UTC+05:30)	-

- 22) Now for auto scaling take ssh of 4 desired instances and increase stress or cpu utilization using following command
- 23) First install epel repository using **amazon-linux-extras install epel -y**

```
[root@ip-172-31-41-250 ~]# amazon-linux-extras install epel -y
```

24) Now install stress package using **yum install stress -y**

```
[root@ip-172-31-41-250 ~]# yum install stress -y
```

25) Now increase stress or cpu utilization using **stress --cpu 8 --io 4 --vm 2 --vm-bytes 128M --timeout 10M &**

```
[root@ip-172-31-41-250 ~]# stress --cpu 8 --io 4 --vm 2 --vm-bytes 128M --timeout 10M &
```

26) Now check stress using top command

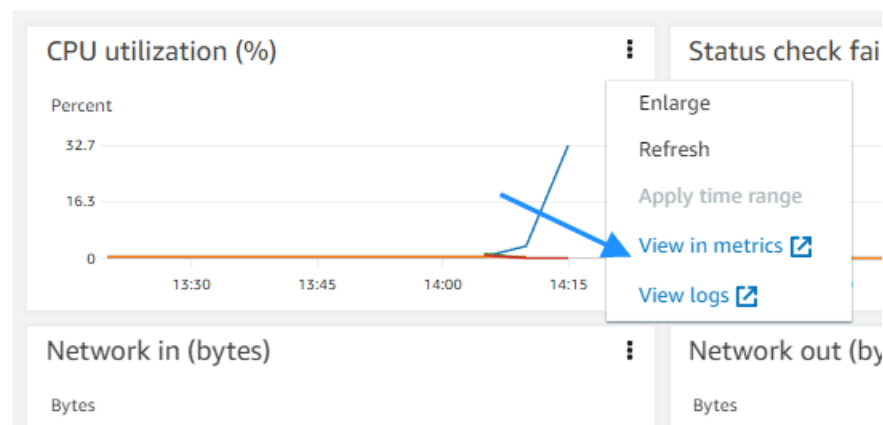
27) Now select all desired instances → monitoring click on 3 dots → and click on view in metrics

**Instances (4/13)** [Info](#)

<input checked="" type="checkbox"/>	Name	Instance ID	Instance state
<input checked="" type="checkbox"/>	home	i-06a9196c361ed79c1	Running
<input checked="" type="checkbox"/>	home	i-03621a595e61d3706	Running
<input type="checkbox"/>	-	i-0b75dd8902162de8c	Terminated
<input checked="" type="checkbox"/>	-	i-082de2f334b7f21c4	Running
<input checked="" type="checkbox"/>	-	i-08589ab2870f869aa	Running
<input type="checkbox"/>	mobile	i-076fa3d2e98e8012c	Running
<input type="checkbox"/>	mobile	i-0160892e10faf1cb3	Running
<input type="checkbox"/>	laptop	i-0fe4c7c69fff1bb56	Running
<input type="checkbox"/>	lanton	i-0d8e9af46e60afe52	Running

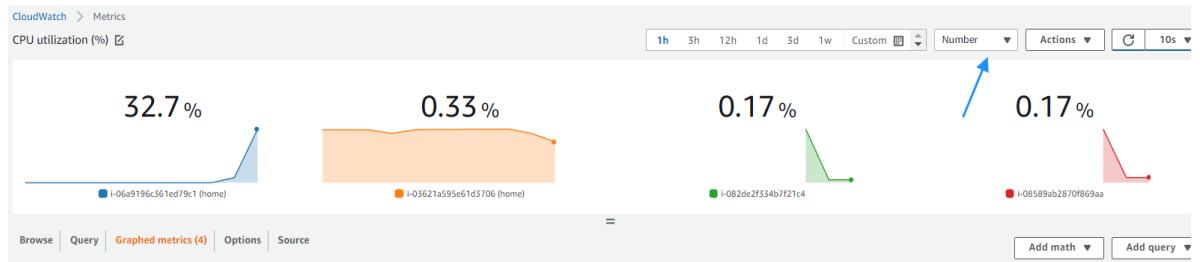
**Instances:** i-06a9196c361ed79c1 (home), i-03621a595e61d3706 (hor

**Monitoring**





28) Now click on numbers it will show cpu utilization in %



29) When cpu utilization goes over 50% it will automatically launch instances to maximum capacity and after hitting url of load balancer DNS it will show 6 public ip which is max capacity of instances