

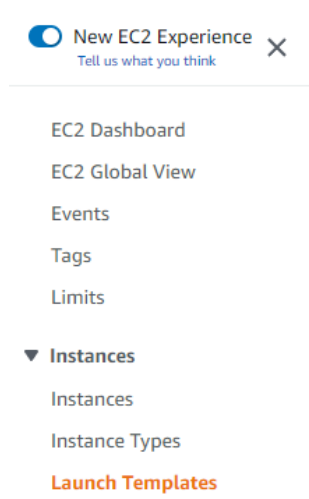
## ASSIGNMENT NO-11

 **Problem Statement:-Build Scaling Plans in AWS that balance load on Different EC2 instances.**

 **Steps:-**

### **Template Creation**

1. Go to EC2 dashboard and in the left side select “Launch Templates”.



2. Click **New launch template** .Give template name, template version, check auto scaling guidance box.

### Create launch template

Creating a launch template allows you to create a saved instance configuration that can be reused, shared and launched at a later time. Templates can have multiple versions.

#### Launch template name and description

Launch template name - *required*

Must be unique to this account. Max 128 chars. No spaces or special characters like '&', '\*', '@'.

Template version description

Max 255 chars

Auto Scaling guidance [Info](#)

Select this if you intend to use this template with EC2 Auto Scaling

☒ Provide guidance to help me set up a template that I can use with EC2 Auto Scaling

► Template tags

► Source template

3. In hardware select ubuntu, instance type t2.micro ,give key pair name(in case if you have existing key pair give that otherwise create new one).

▼ Instance type
Info
Advanced

Instance type

t2.micro

Free tier eligible

Family: t2
1 vCPU
1 GiB Memory
Current generation: true
On-Demand Linux pricing: 0.0124 USD per Hour
On-Demand Windows pricing: 0.017 USD per Hour
On-Demand RHEL pricing: 0.0724 USD per Hour
On-Demand SUSE pricing: 0.0124 USD per Hour

All generations

[Compare instance types](#)

▼ Key pair (login)
Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name

jhantu007

Create new key pair

- In Network settings select existing security group and select the existed security group which is created previously.

▼ Network settings
Info

Subnet Info

Don't include in launch template

Create new subnet

When you specify a subnet, a network interface is automatically added to your template.

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Select existing security group

Create security group

Security groups Info

Select security groups

sw sg-033ca2fc3fc5ab788
VPC: vpc-09a60d23ab93463d1

Compare security group rules

► Advanced network configuration

- In advanced details in user data section type the following commands-

#### User data - optional Info

Enter user data in the field.

```
#!/bin/bash
apt-get update
apt-get install -y nginx
systemctl start nginx
systemctl enable nginx
apt-get install -y git
curl -sL https://deb.nodesource.com/setup_18.x | sudo -E bash -
apt-get install -y nodejs
git clone https://github.com/manas003884/repo2.git
cd repo2
npm install
node index.js
```

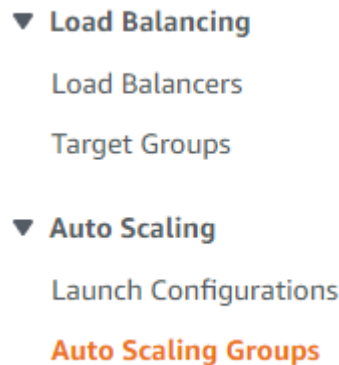
(In git clone line user can give his repo name and cd line the repo name will be according to his repo name.)

Note- before copying the github repo make sure it is public. If not then perform following steps-

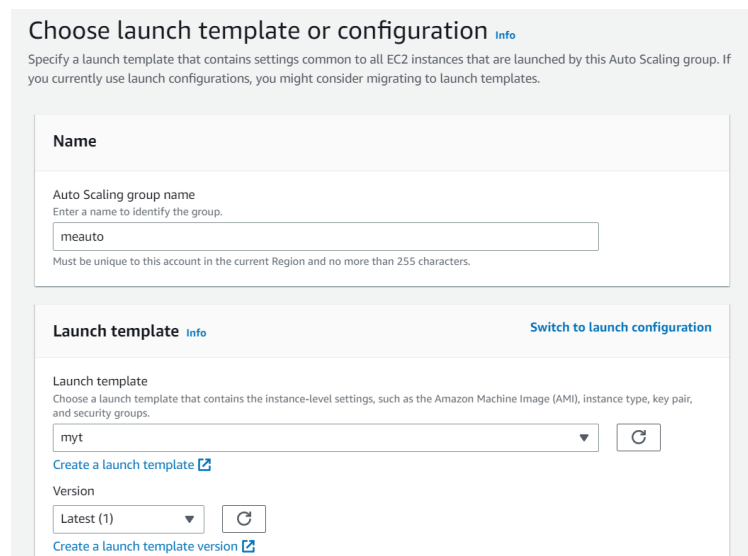
- a. Go to repo settings and by scrolling down in danger zone click “**change repository visibility**”. And change it to public.

## Auto Scaling Groups

1. In EC2 dashboard click Auto Scaling Groups. Click on **Create Auto Scaling group**.



2. Give auto scaling group name (ex-manasautoscale1). In launch template click on the existing template(ex-mantemplate1) ,give version **Latest(1)**and click next



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.

meauto

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

**Launch template** [Info](#) [Switch to launch configuration](#)

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.

myt

[Create a launch template](#)

Version

Latest (1)

[Create a launch template version](#)

3. In Network ,Availability Zones click all the zones and subnets and click 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-09a60d23ab93463d1  
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-0bed1f90424dab9b6 X  
172.31.32.0/20 Default

ap-south-1b | subnet-08ea2f92cc33219d3 X  
172.31.0.0/20 Default

ap-south-1c | subnet-0619bbd98a4ff73c9 X  
172.31.16.0/20 Default

[Create a subnet](#)

4. in load balancing click “**Attach to a new load balancer**”, in load balancer scheme select “**internet-facing**”, in listeners and routing give port number 4000 and default routing select autoscaling group(ex-manautoscaling1-1|HTTP). And click next.

## 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.

### Load balancer scheme

Scheme cannot be changed after the load balancer is created.

☐ Internal

☒ Internet-facing

### Listeners and routing

#### Listeners and routing

If you require secure listeners, or multiple listeners, you can configure them from the [Load Balancing console](#) after your load balancer is created.

Protocol	Port	Default routing (forward to)
HTTP	4000	jhauto-1   HTTP

#### Tags - optional

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

5. in **Group size** give desired capacity 2,minimum capacity 2,maximum capacity 3.

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

Specify the size of the Auto Scaling group maximum capacity limits. Your desired

Desired capacity

Minimum capacity

Maximum capacity

6. In Scaling policies click “Target tracking scaling policy” and instances need section type 300.

### Scaling policies - optional

Choose whether to use a scaling policy to dynamically resize your Auto Scaling 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 need

seconds warm up before including in metric

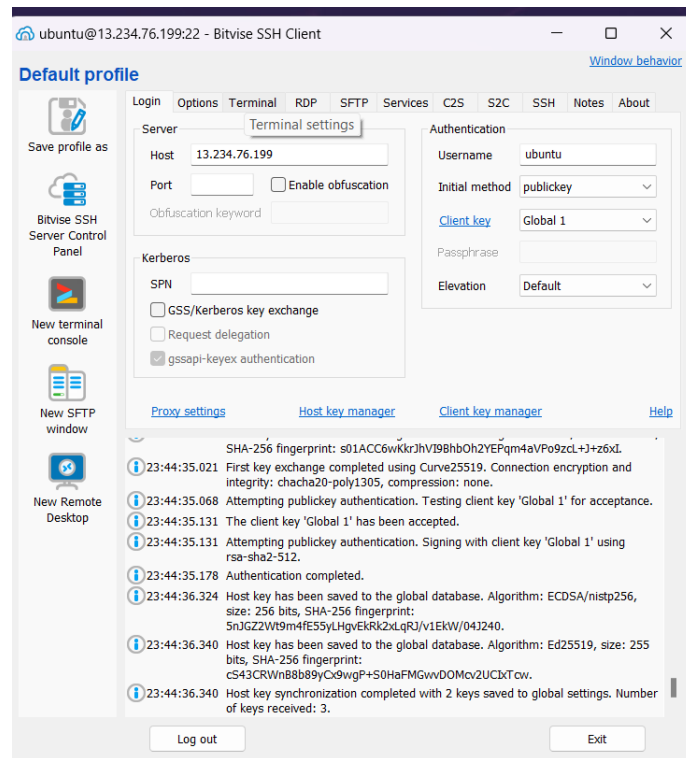
**Click next** and click **Create Auto Scaling group**. And it will be created.

Now, we have to crash these two running servers. for that we will crash one server with bitvisessh client and in another one we will crash through directly opening terminal.



#### For one server:

1. Copy public IPv4 address(ex-65.2.3.43) and paste it on Bitvise SSH Client. Give username ubuntu, initial method publickey, in client key manager import that same existed key pair .pem file(ex-key003.pem) and click Global1 in Client key.and click log in.



2. In Terminal type **nano infi1.sh** and in the file write the following lines of code and save it.

```
GNU nano 6.2
#!/bin/bash

while true
do
    echo "Looping forever"
    # Add other commands to run in the loop here
done
```

3. To execute the file give command **chmod +x infi1.sh**.
4. To run give command **./infi1.sh** and infinite loop will start.

```
Looping forever
Looping forever
Looping forever
Looping forever
Looping forever
Looping forever
Looping forever
Looping forever
Looping forever
Looping forever
Looping forever
Looping forever
Looping forever
Looping forever
Looping forever
Looping forever
Looping forever
```

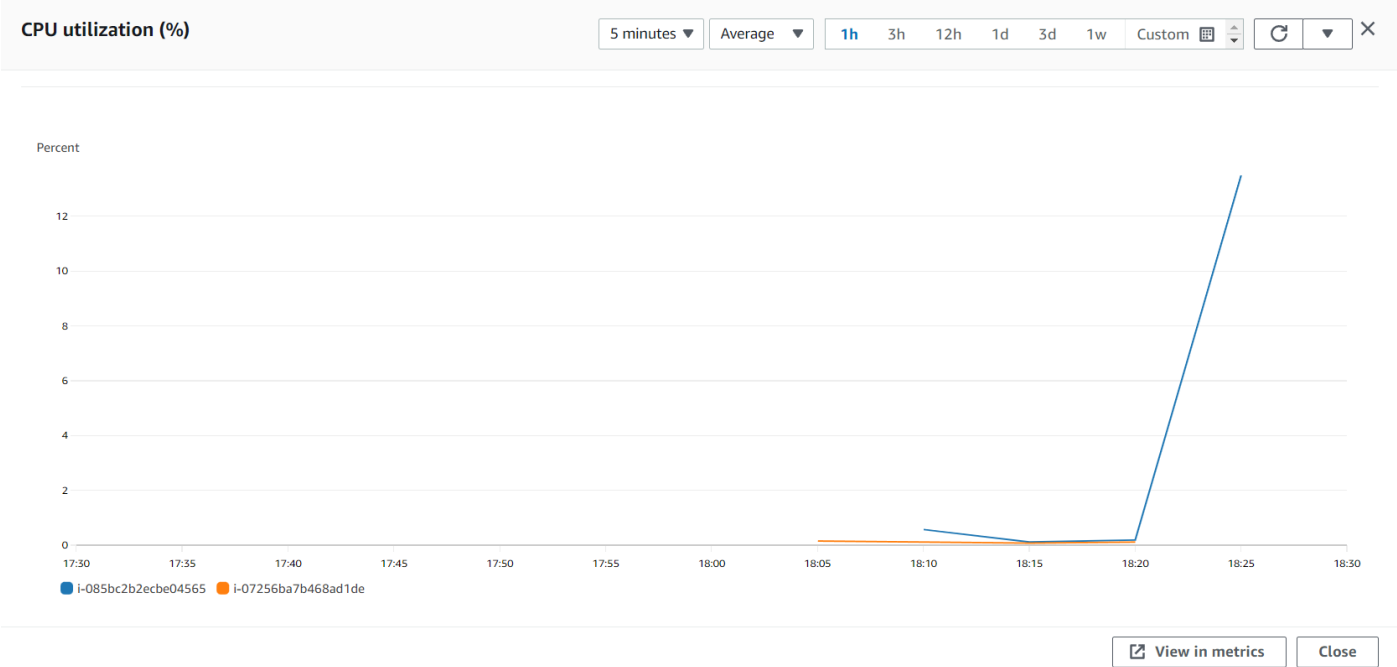
 **For another server:**

1. Click on connect option and one terminal will open.
2. In the terminal type the command as same as previous. And run .

```
To run a command as administrator (user "root"),
See "man sudo_root" for details.

ubuntu@ip-172-31-35-242:~$ nano infi2.sh
ubuntu@ip-172-31-35-242:~$ chmod +x infi2.sh
ubuntu@ip-172-31-35-242:~$ ./infi2.sh
```

Now, the servers will be overloaded and we can see that by click on CPU utilization.



After some time, we can see that a new instance is created automatically for load balancing.

Instances (3) <span>Info</span>								
<div>Find instance by attribute or tag (case-sensitive)</div>								
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
<input type="checkbox"/>	-	i-085bc2b2ecbe04565	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1b	ec2-13-234-
<input type="checkbox"/>	cr1	i-08c9eee720ee902eb	Terminated	t2.micro	-	No alarms	ap-south-1a	-