

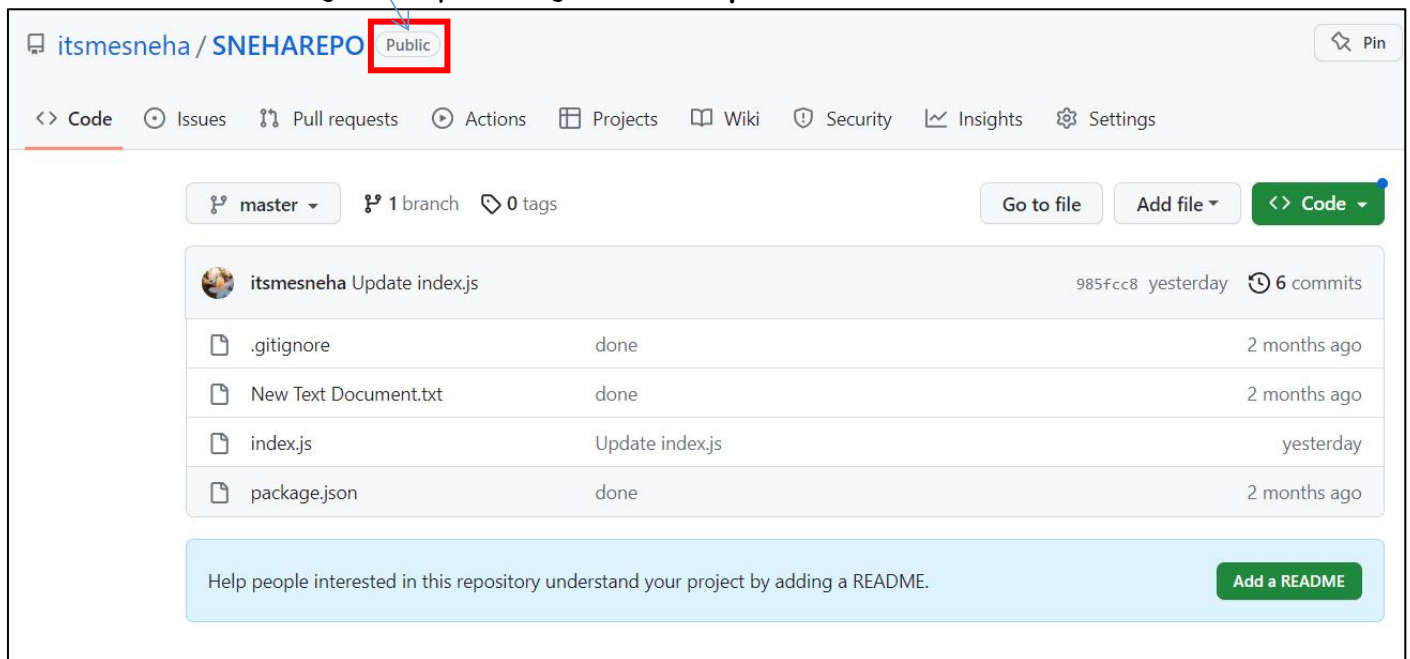
Assignment: 10

Title : Deploy a project from github to EC2 by creating new security group and user data.

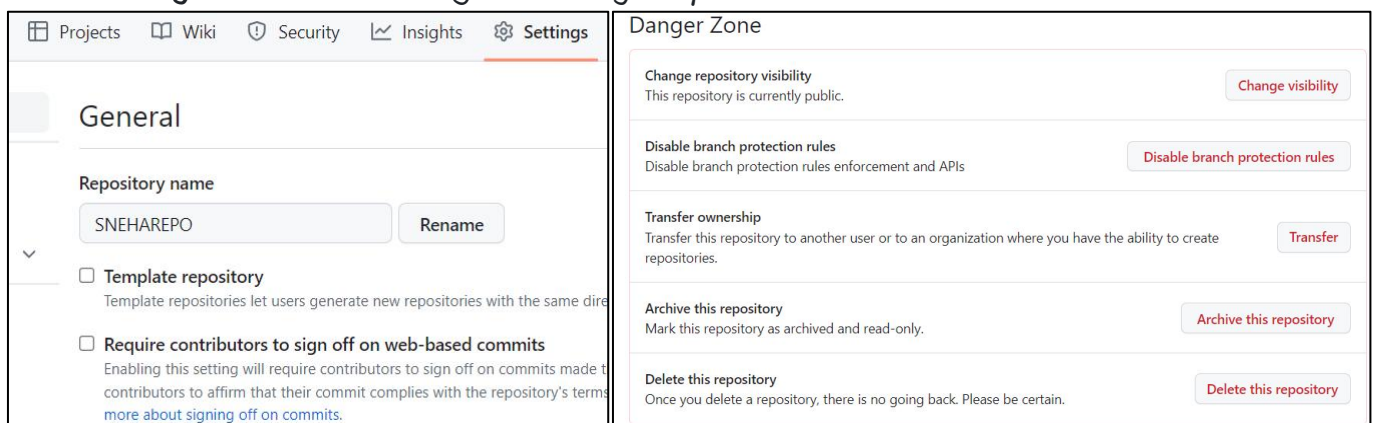
About :

A security group acts as a virtual firewall to control inbound and outbound traffic.

- ◆ First open your github account , go to your repository that was already created in previous assignments - <https://github.com/itsmesneha/SNEHAREPO>
- ◆ Make sure that your repository is made 'public'



- ◆ If the repository is not public, go to **Settings** -> scroll down to **Change repository visibility** and make change visibility to public.



Steps to create a security group :

1. Go to EC2 Dashboard -> Security Groups
2. Delete all security groups (default cannot be deleted) -
Select the security group , go to actions - choose Delete Security groups

3. Click on *Create Security Group* to create a new security group.

Security Groups (1/2) [Info](#)

Filter security groups

	Name	Security group ID	Security group name	Description	Owner
<input checked="" type="checkbox"/>	-	sg-011957412b00b9d5d	snehasecurity	snehasecurity	044915352154
<input type="checkbox"/>	-	sg-0992ffa5c891aa71d	default	default VPC security gr...	044915352154

4. Now you need to give the name (*SnehaSecurityGroup*) , description details and change the inbound rules as follows :

Click on *Add rule* and then complete the following changes :

Type	Protocol	Port range	Source
SSH	TCP	22	Anywhere / 0.0.0.0
HTTP	TCP	80	Anywhere / 0.0.0.0
HTTPS	TCP	443	Anywhere / 0.0.0.0
Common TCP	TCP	0	Anywhere / 0.0.0.0

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name [Info](#)

Name cannot be edited after creation.

Description [Info](#)

VPC [Info](#)

Inbound rules [Info](#)

Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info	
SSH	TCP	22	Anywh... <input type="text" value="0.0.0.0/0"/>		Delete
HTTP	TCP	80	Anywh... <input type="text" value="0.0.0.0/0"/>		Delete
HTTPS	TCP	443	Anywh... <input type="text" value="0.0.0.0/0"/>		Delete
Custom TCP	TCP	0	Anywh... <input type="text" value="0.0.0.0/0"/>		Delete

Add rule

Security group (sg-086781f3b63bfd85b | SnehaSecurityGrp) was created successfully

Details

EC2 > Security Groups > sg-086781f3b63bfd85b - SnehaSecurityGrp

sg-086781f3b63bfd85b - SnehaSecurityGrp

Details			
Security group name SnehaSecurityGrp	Security group ID sg-086781f3b63bfd85b	Description SnehaSecurityGrp	VPC ID vpc-02a7beb0499034079
Owner 044915352154	Inbound rules count 4 Permission entries	Outbound rules count 1 Permission entry	

5. Click on create Security group and your security group is created.

Steps to create an instance using security group :

1. Click on **EC2 dashboard** and choose **Launch instance**
2. In the **launch an Instance** page give the name e.g. **myinstance21**
3. Under **Quick start** , choose **Ubuntu** (eligible for free tier).

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name
myinstance21 [Add additional tags](#)

Quick Start

Amazon Linux macOS **Ubuntu** Windows Red Hat

aws Mac ubuntu Microsoft Red Hat

Amazon Machine Image (AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type
ami-007855ac798b5175e (64-bit (x86)) / ami-0c6c29c5125214c77 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

4. In **Instance type** select **t2 micro** and under **Key pair(login)** and select the key pair you have already created (**snehaa1234**) or create a new key pair by clicking on **Create new key pair** and create a new one.

Instance type Info

Instance type

t2.micro Free tier eligible

Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Windows pricing: 0.0162 USD per Hour
On-Demand SUSE pricing: 0.0116 USD per Hour
On-Demand RHEL pricing: 0.0716 USD per Hour
On-Demand Linux pricing: 0.0116 USD per Hour

Key pair (login) Info

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

Key pair name - required

snehaa1234

5. Now , under **Network settings** choose the **Security group** you previously created.

Network settings Info Edit

Network Info
vpc-02a7beb0499034079

Subnet Info
No preference (Default subnet in any availability zone)

Auto-assign public IP Info
Enable

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.

☐ Create security group ☒ Select existing security group

Security groups Info
Select security groups

SnehaSecurityGrp sg-086781f3b63bfd85b X
VPC: vpc-02a7beb0499034079 [Compare security group rules](#)

6. Under the **user data** , we need to copy paste the code given and change the git clone path to the one with our repository
And include the repository name beside **cd** command



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/itsmesneha/SNEHAREPO.git
cd SNEHAREPO
npm install
node index.js
```

```
#!/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/.../...
cd ...
npm install
node index.js
```

Now the instance is created as seen below :

Instances (1) Info			
<input type="text" value="Find instance by attribute or tag (case-sensitive)"/>			
<input type="checkbox"/>	Name	Instance ID	Instance state
<input type="checkbox"/>	myinstance21	i-0b0649233902b8187	Running

7. Click on the **instance id** which will lead to the **instance summary** of the instance

8. Copy the **public ipv4 address**

EC2 > Instances > i-0b0649233902b8187

Instance summary for i-0b0649233902b8187 (myinstance21) [Info](#)

Updated less than a minute ago

Instance ID

i-0b0649233902b8187 (myinstance21)

IPv6 address

—

Hostname type

IP name: ip-172-31-29-100.ec2.internal

Answer private resource DNS name

IPv4 (A)

Auto-assigned IP address

184.72.212.12 [Public IP]

Public IPv4 address copied

Public IPv4 address

184.72.212.12 | [open address](#)

Instance state

Running

Private IP DNS name (IPv4 only)

ip-172-31-29-100.ec2.internal

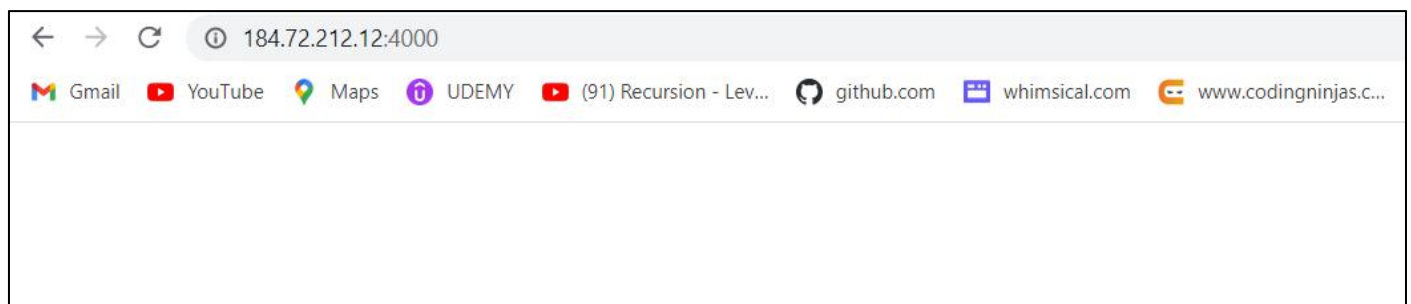
Instance type

t2.micro

VPC ID

vpc-02a7beb0499034079

Paste the copied address in the browser along with the port number and see it runs.



Assignment: 11

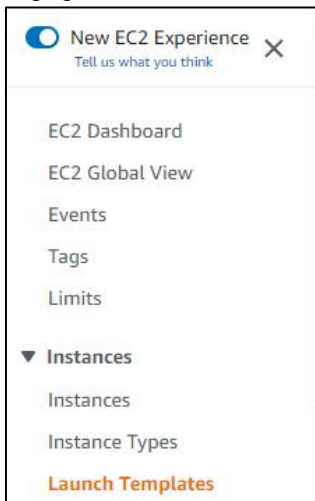
Title : Build Scaling Plans in AWS that balance load on different EC2 instances.

About:

Elastic Load Balancing automatically distributes your incoming application traffic across all the EC2 instances that you are running. Elastic Load Balancing helps to manage incoming requests by optimally routing traffic so that no one instance is overwhelmed.

Steps to create Template:

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 re-used 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

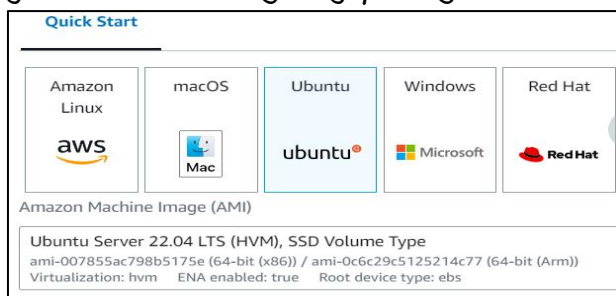
Max 255 chars

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

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](#)

Instance type

t2.micro Free tier eligible

Family: t2 1 vCPU 1 GiB Memory Current generation: true

On-Demand Windows pricing: 0.0162 USD per Hour

On-Demand SUSE pricing: 0.0116 USD per Hour

On-Demand RHEL pricing: 0.0716 USD per Hour

On-Demand Linux pricing: 0.0116 USD per Hour

▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have an existing key pair or create a new one.

Key pair name - *required*

4. Now , under Network settings choose the Security group you previously created.

Network settings [Info](#) [Edit](#)

Network [Info](#)
vpc-02a7beb0499034079

Subnet [Info](#)
No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)
Enable

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.

☐ Create security group ☒ Select existing security group

Security groups [Info](#)
Select security groups

SnehaSecurityGrp sg-086781f3b63bfd85b X [Compare security group rules](#)

VPC: vpc-02a7beb0499034079

5. Under the user data, we need to copy paste the code given and change the git clone path to the one with our repository. And include the repository name beside cd command



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/itsmesneha/SNEHAREPO.git
cd SNEHAREPO
npm install
node index.js
```

```
#!/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/.../...
cd ...
npm install
node index.js
```

Note - before copying the github repo make sure it is public. If not then perform following steps - Go to repo settings and by scrolling down in danger zone click "change repository visibility". And change it to public.

Steps to create Autoscaling groups:

1. In EC2 dashboard click Auto Scaling Groups. Click on Create Auto Scaling group.
2. Give auto scaling group name (ex-myautoscale1). In launch template click on the existing template(ex-mytemplate1), give version Latest(1) and click next

Load Balancing

Load Balancers

Target Groups

Auto Scaling

Launch Configurations

Auto Scaling Groups

Name

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

myautoscale1

Must be unique to this account in the current Region and no more than 64 characters.

Launch template [Info](#)

Launch template
Choose a launch template that contains the instance-level settings, such as the AMI, instance profile, and security groups.

mantemplate1

[Create a launch template](#)

Version

Latest (1) [Refresh](#)

[Create a launch template version](#)

3. In **Network**, **Availability Zones and subnets** click all the zones and click next.

Network Info

For most applications, you can use multiple Availability Zones and subnets. The default VPC and default subnets are suitable for getting started.

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

vpc-044ae7b2b99754d9f
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 your VPC.

Select Availability Zones and subnets

ap-south-1a | subnet-0c5794505f6771b54
172.31.32.0/20 Default

ap-south-1b | subnet-0f1387f5dd527adc8
172.31.0.0/20 Default

ap-south-1c | subnet-0b96504bd63890dea
172.31.16.0/20 Default

8. 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
If you require secure listeners, or multiple listeners, you can configure them from the [Load Balancing console](#) after your load balancer is created.

Protocol: HTTP Port: 4000 Default routing (forward to): manautoscaling1-1 | HTTP

9. In **Group size** give desired capacity 2, minimum capacity 2, maximum capacity 3.

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

Group size - optional Info

Specify the size of the Auto Scaling group and its minimum and maximum capacity limits. Your desired capacity must be between the minimum and maximum capacity limits.

Desired capacity
2

Minimum capacity
2

Maximum capacity
3

Scaling policies - optional

Choose whether to use a scaling policy to dynamically resize your Auto Scaling group based on 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
Target Tracking Policy

Metric type
Average CPU utilization

Target value
50

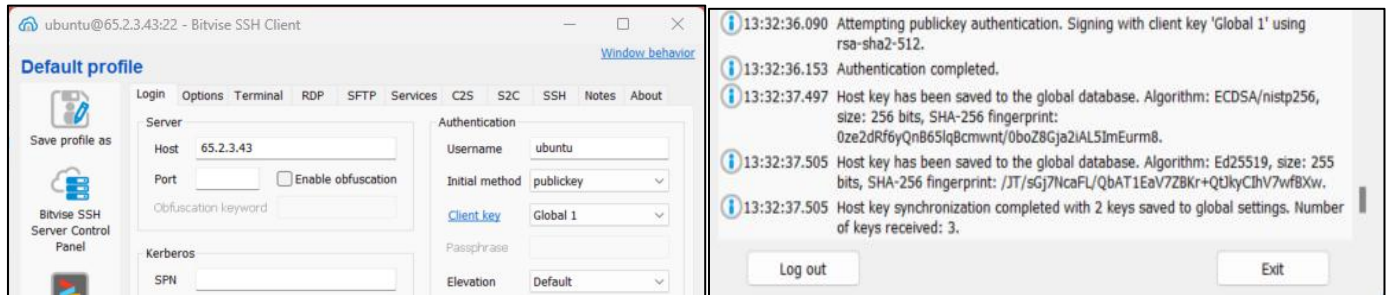
Instances need
300 seconds warm up before including in metric

11. 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 bitvise ssh 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.
- 2.Give username ubuntu, initial method publickey, in client key manager import that same existed key pair .pem file(ex-snehaa1234.pem) and click Global1 in Client key.and click log in. We are already logged in now .



- 3.In Terminal type nano infi1.sh and in the file write the following lines of code and save it. To execute the file give command chmod +x infi1.sh.
4. To run give command ./infi1.sh and infinite loop will start.

The code :

```
#!/bin/bash
```

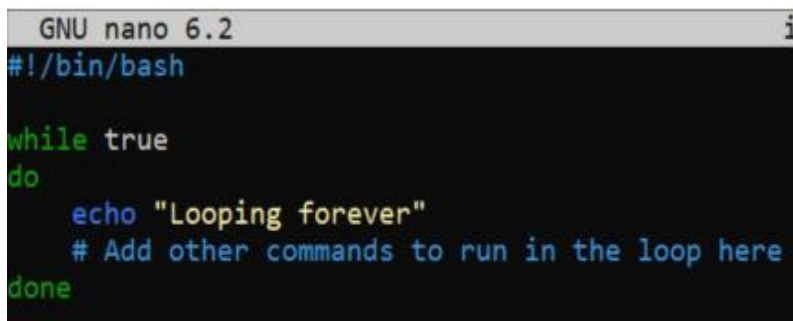
```
while true
```

```
do
```

```
    echo "Looping forever"
```

```
    # Add another commands to run in the loop here
```

```
done
```



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 .

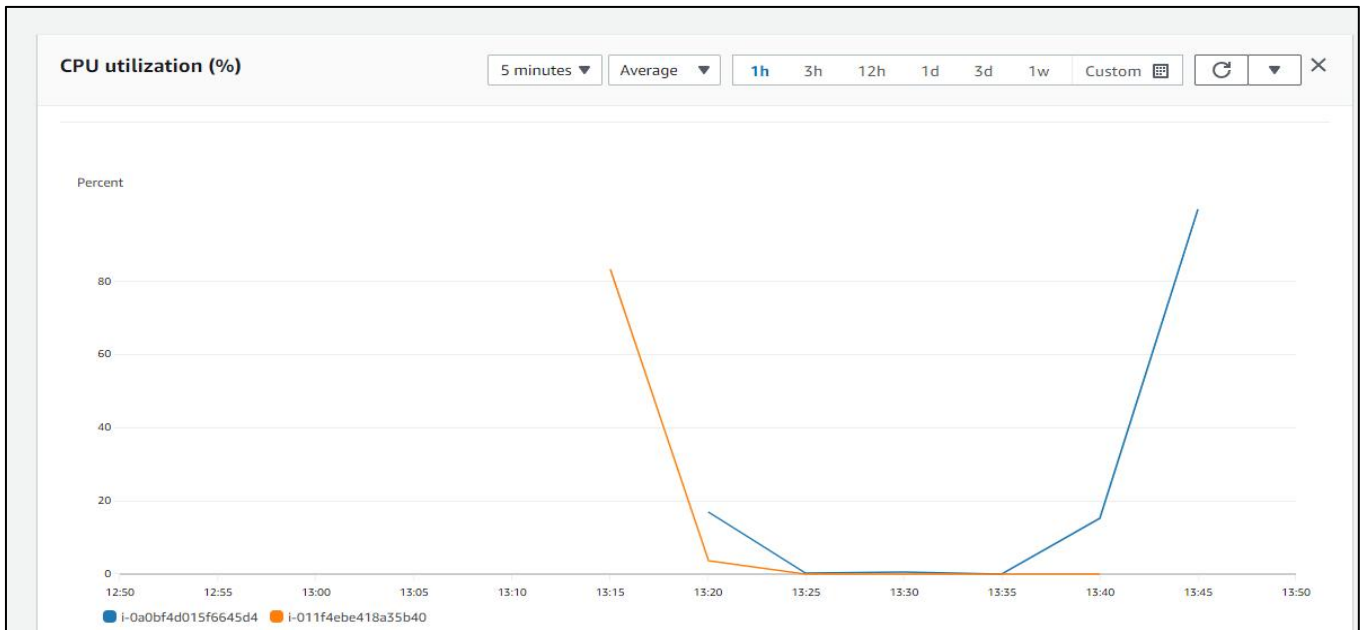

```

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 new instance is created automatically for load balancing.

Instances (4) Info					Connect	Instance state
Find instance by attribute or tag (case-sensitive)						
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type		
<input type="checkbox"/>	manec2	i-09cc62e14b3f1c3d1	Running	t2.micro		
<input type="checkbox"/>	-	i-0a0bf4d015f6645d4	Running	t2.micro		
<input type="checkbox"/>	-	i-020587bf58cb18c94	Running	t2.micro		
<input type="checkbox"/>	-	i-011f4ebe418a35b40	Running	t2.micro		