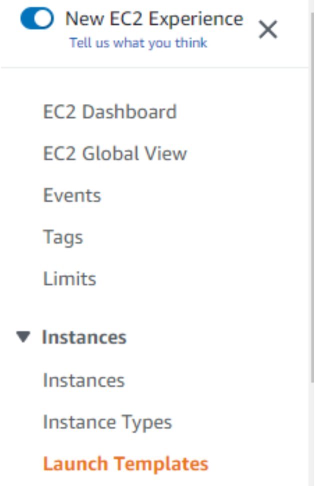
ASSIGNMENT NO-11

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

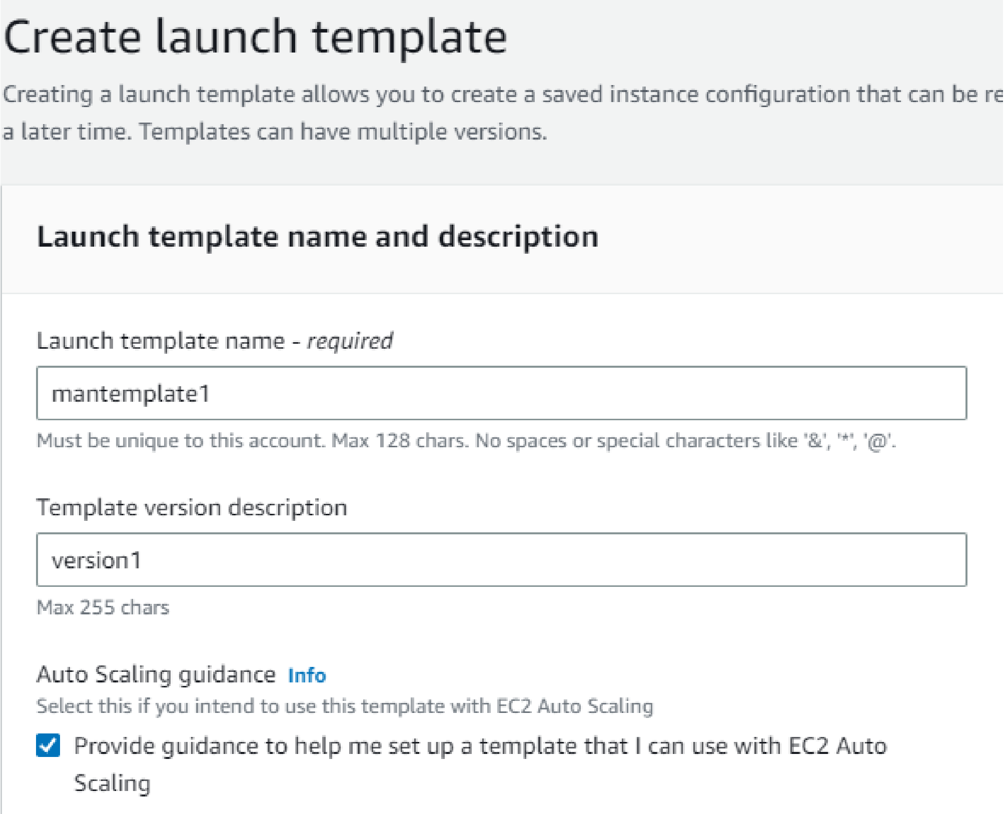
🟩 **Steps:-**

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 re a later time. Templates can have multiple versions.

Launch template name and description

Launch template name - *required*

mantemplate1

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

Template version description

version1

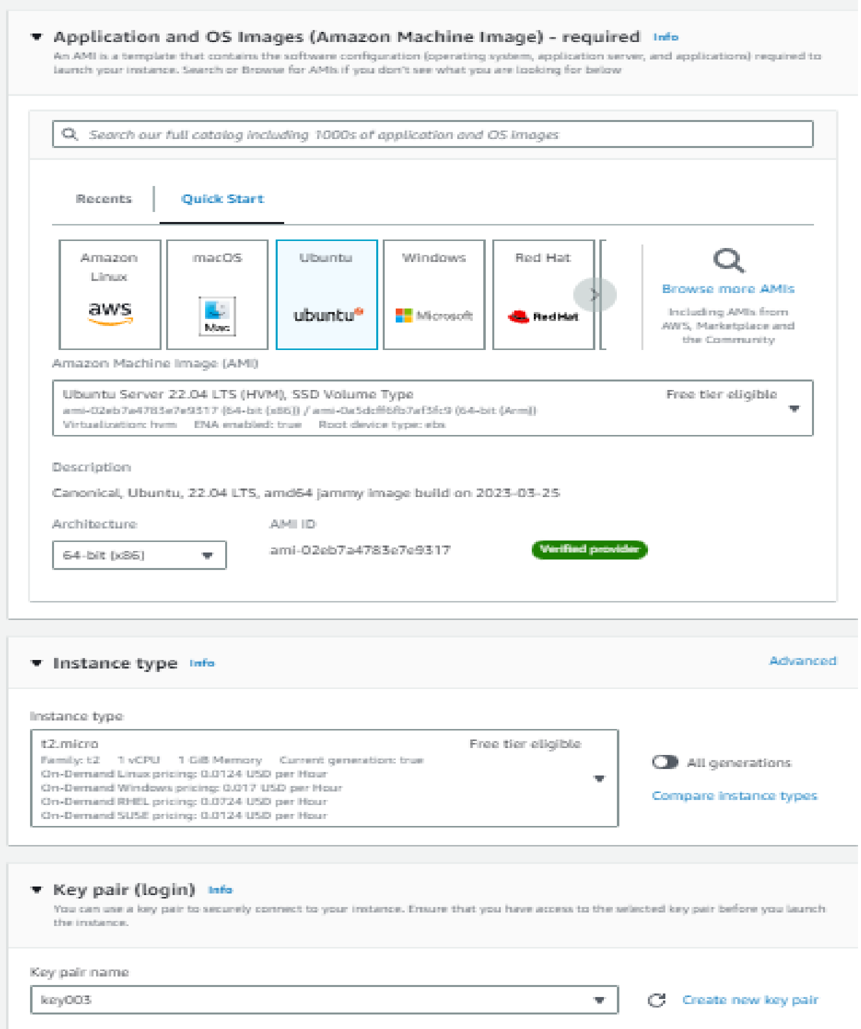
Max 255 chars

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).



Application and OS Images (Amazon Machine Image) - required Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Recents | Quick Start

Amazon Linux | macOS | Ubuntu | Windows | Red Hat

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type — Free tier eligible

ami-02eb7a4783e7e9317 (64-bit (x86)) / ami-0a5dcf6fb7af3fc9 (64-bit (Arm))

Virtualization: hvm   ENA enabled: true   Root device type: ebs

Description

Canonical, Ubuntu, 22.04 LTS, amd64 jammy image build on 2023-03-25

Architecture: 64-bit (x86)

AMI ID: ami-02eb7a4783e7e9317 — Verified provider

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

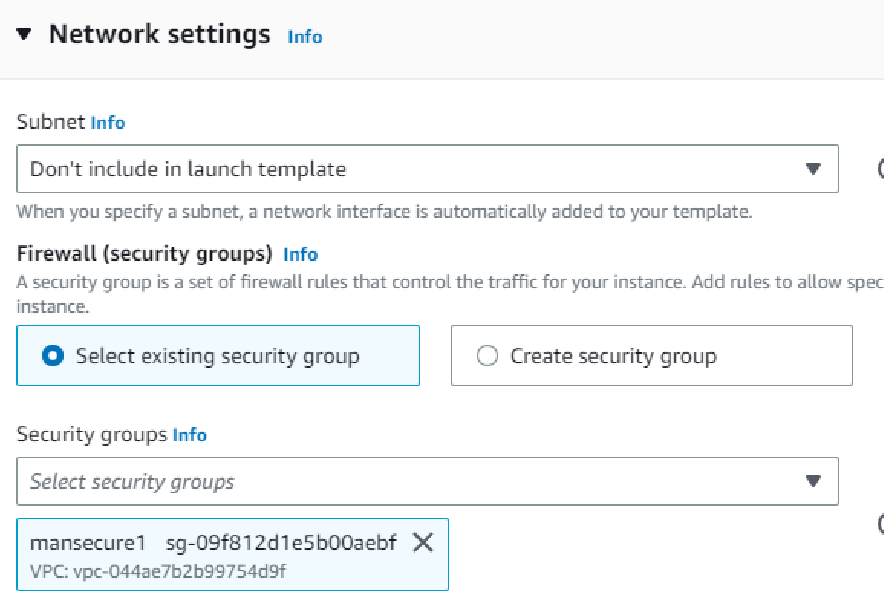
Key pair (login) Info

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: key003

Create new key pair

4. 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

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 spec instance.

● Select existing security group   ○ Create security group

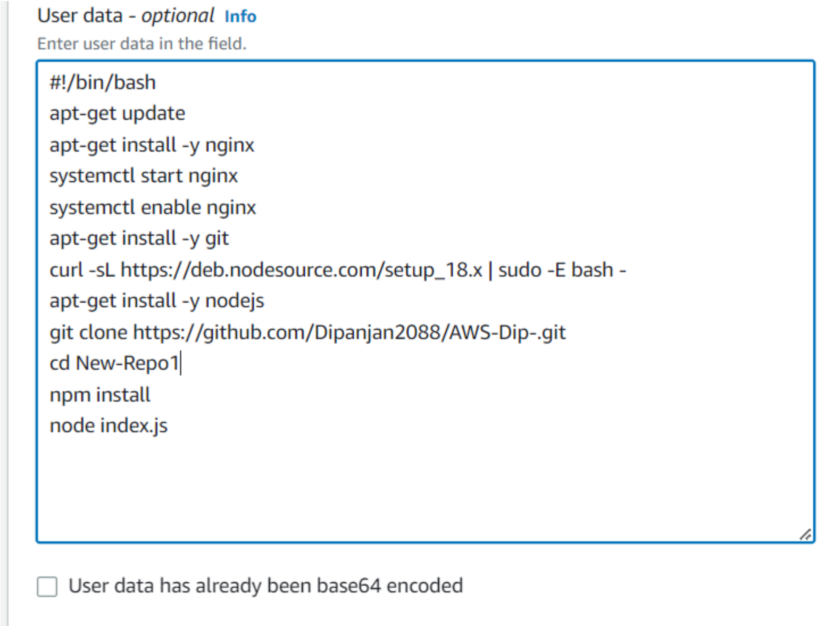
Security groups Info

Select security groups

mansecure1   sg-09f812d1e5b00aebf

VPC: vpc-044ae7b2b99754d9f

5. 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/Dipanjan2088/AWS-Dip-.git
cd New-Repo1
npm install
node index.js
```

□ User data has already been base64 encoded

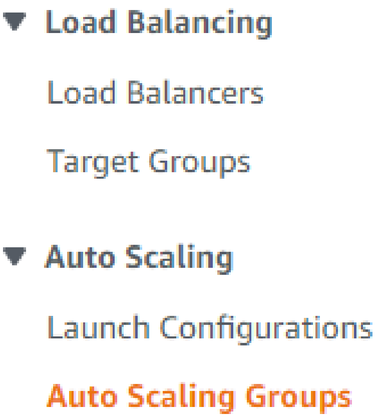
(In git clone line user can give his repo name and cd line the repo name will be according to his repo name.)

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

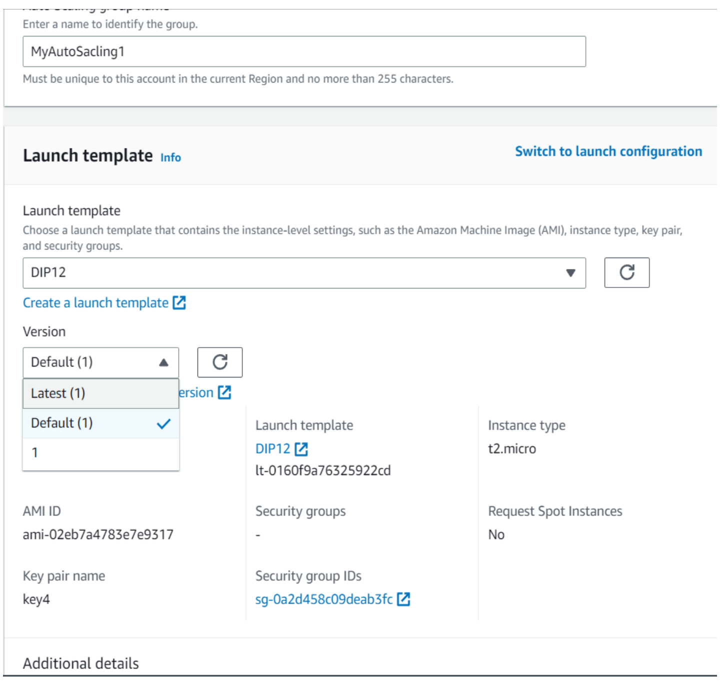
a. Go to repo settings and by scrolling down in danger zone click “**change repository visibility”.** And change it to public.

Auto Scaling Groups

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



2. Give auto scaling group name (ex-MyAutoSacing1). In launch template click on the existing template(ex-DIP1) ,give version **Latest(1)** and click next



Enter a name to identify the group.

MyAutoSacling1

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.

DIP12

Create a launch template

Version

Default (1)

Latest (1)

Default (1) ✓

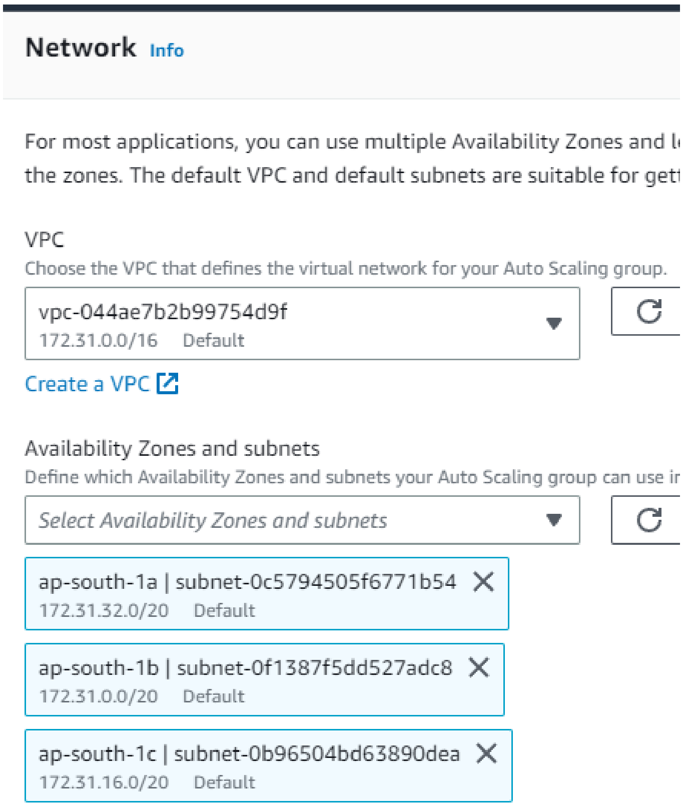
1

ersion

	Launch template DIP12 lt-0160f9a76325922cd	Instance type t2.micro
AMI ID ami-02eb7a4783e7e9317	Security groups -	Request Spot Instances No
Key pair name key4	Security group IDs sg-0a2d458c09deab3fc	

Additional details

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 l
the zones. The default VPC and default subnets are suitable for gett

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 ir

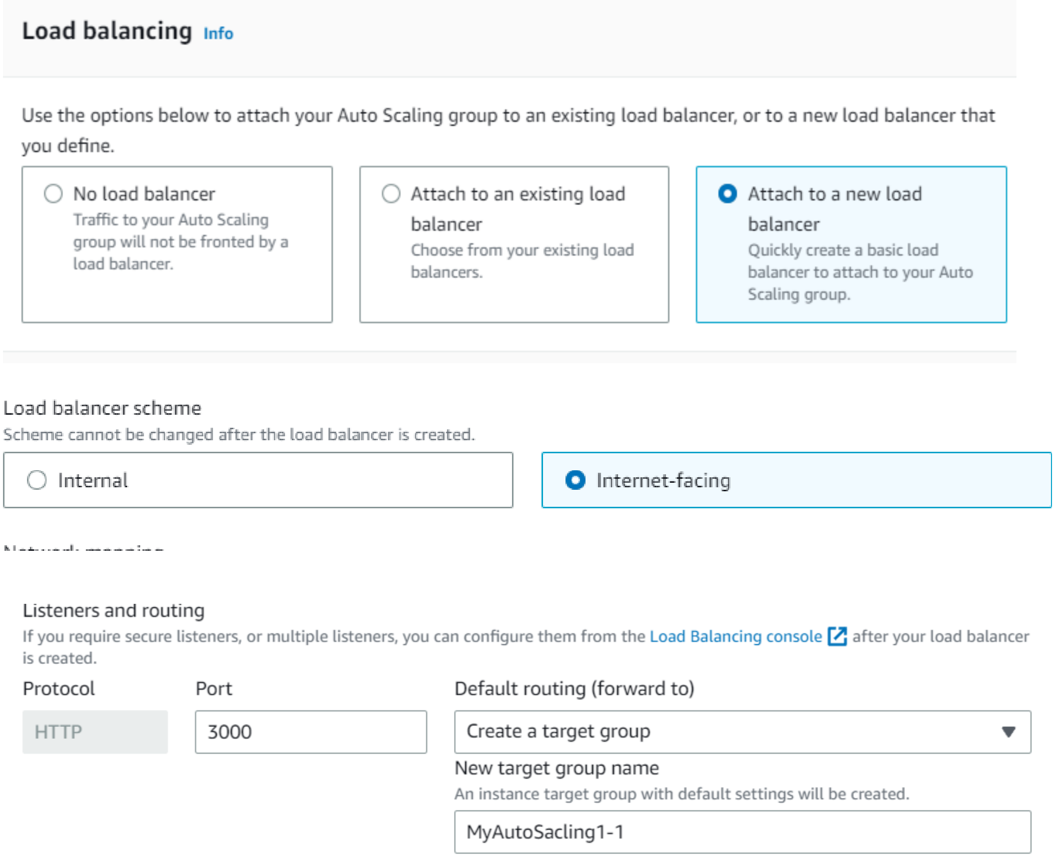
Select Availability Zones and subnets

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

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 3000 and default routing select autoscaling group(ex-MyAutoscaling1-1|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

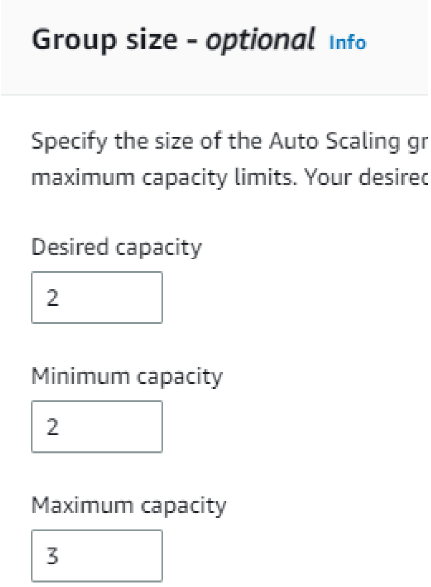
● Internet-facing

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	Port	Default routing (forward to)
HTTP	3000	Create a target group

New target group name
An instance target group with default settings will be created.
MyAutoSacling1-1

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



Group size - *optional* Info

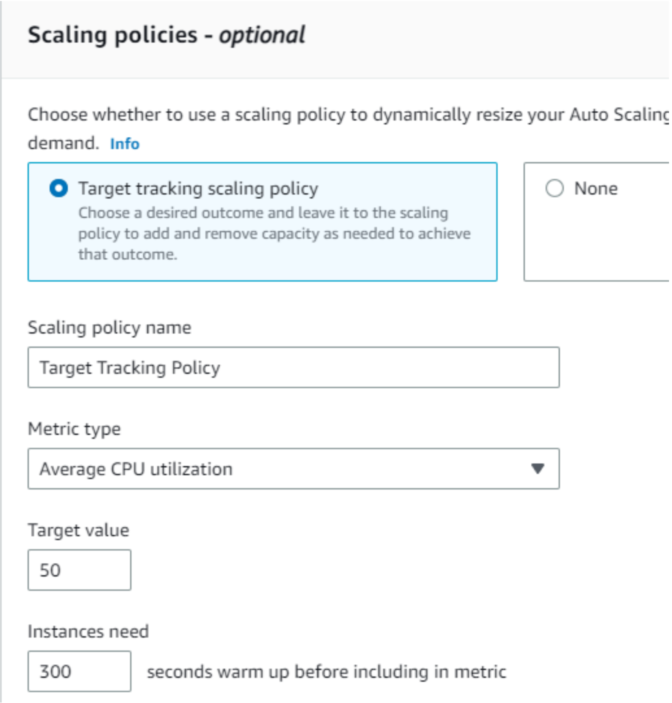
Specify the size of the Auto Scaling gr
maximum capacity limits. Your desirec

Desired capacity
2

Minimum capacity
2

Maximum capacity
3

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: Target Tracking Policy

Metric type: Average CPU utilization

Target value: 50

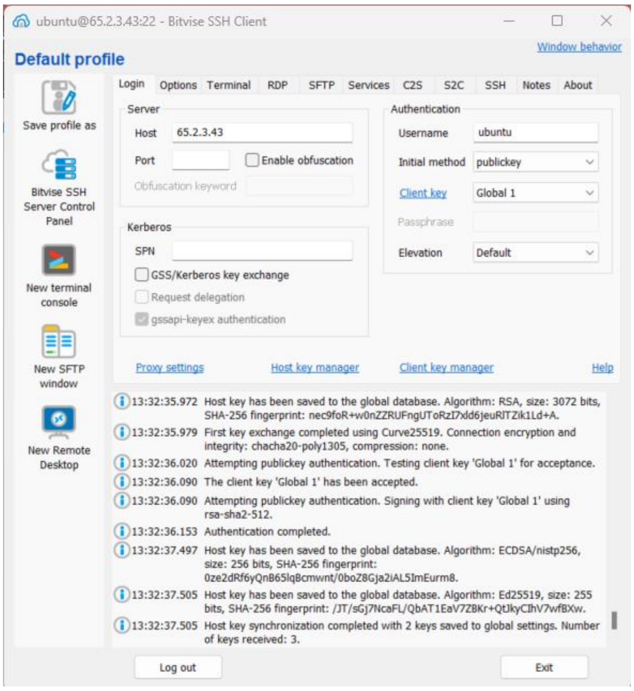
Instances need: 300 seconds warm up before including in metric

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 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. 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.



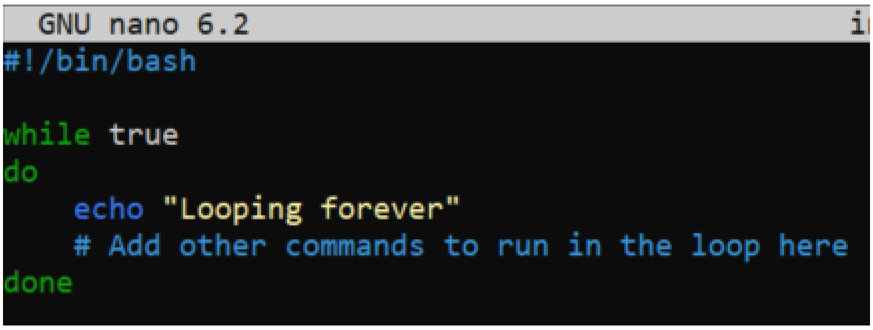
Server
Host: 65.2.3.43
Port:
□ Enable obfuscation
Obfuscation keyword:

Kerberos
SPN:
□ GSS/Kerberos key exchange
□ Request delegation
☑ gssapi-keyex authentication

Authentication
Username: ubuntu
Initial method: publickey
Client key: Global 1
Passphrase:
Elevation: Default

```
13:32:35.972 Host key has been saved to the global database. Algorithm: RSA, size: 3072 bits,
             SHA-256 fingerprint: nec9foR+w0nZZRUFngUToRzI7xld6jeuRlTZik1Ld+A.
13:32:35.979 First key exchange completed using Curve25519. Connection encryption and
             integrity: chacha20-poly1305, compression: none.
13:32:36.020 Attempting publickey authentication. Testing client key 'Global 1' for acceptance.
13:32:36.090 The client key 'Global 1' has been accepted.
13:32:36.090 Attempting publickey authentication. Signing with client key 'Global 1' using
             rsa-sha2-512.
13:32:36.153 Authentication completed.
13:32:37.497 Host key has been saved to the global database. Algorithm: ECDSA/nistp256,
             size: 256 bits, SHA-256 fingerprint:
             0ze2dRf6yQnB65lqBcmwnt/0boZ8Gja2iAL5ImEurm8.
13:32:37.505 Host key has been saved to the global database. Algorithm: Ed25519, size: 255
             bits, SHA-256 fingerprint: /JT/sGj7NcaFL/QbAT1EaV7ZBKr+QtJkyCIhV7wfBXw.
13:32:37.505 Host key synchronization completed with 2 keys saved to global settings. Number
             of keys received: 3.
```

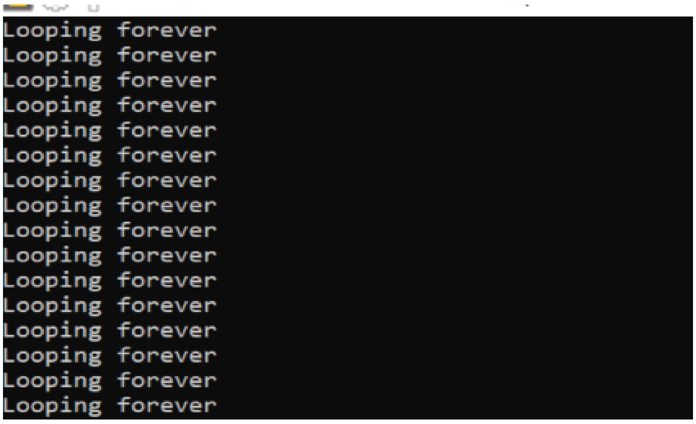
2. In Terminal type **nano infi1.sh** and in the file write the following lines of code and save it.



```
#!/bin/bash

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

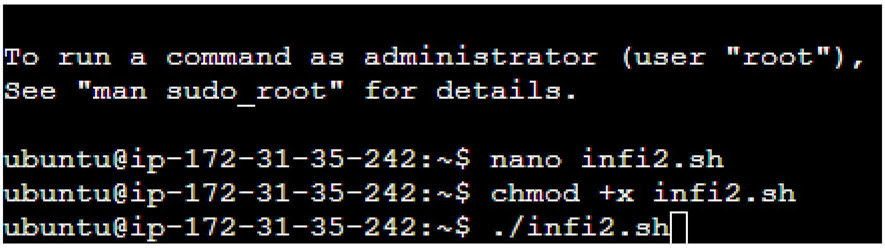
3. To execute the file give command **chmod +x infi1.sh**.
4. To run give command **./infi1.sh** and infinite loop will start.



```
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
```

For another server:

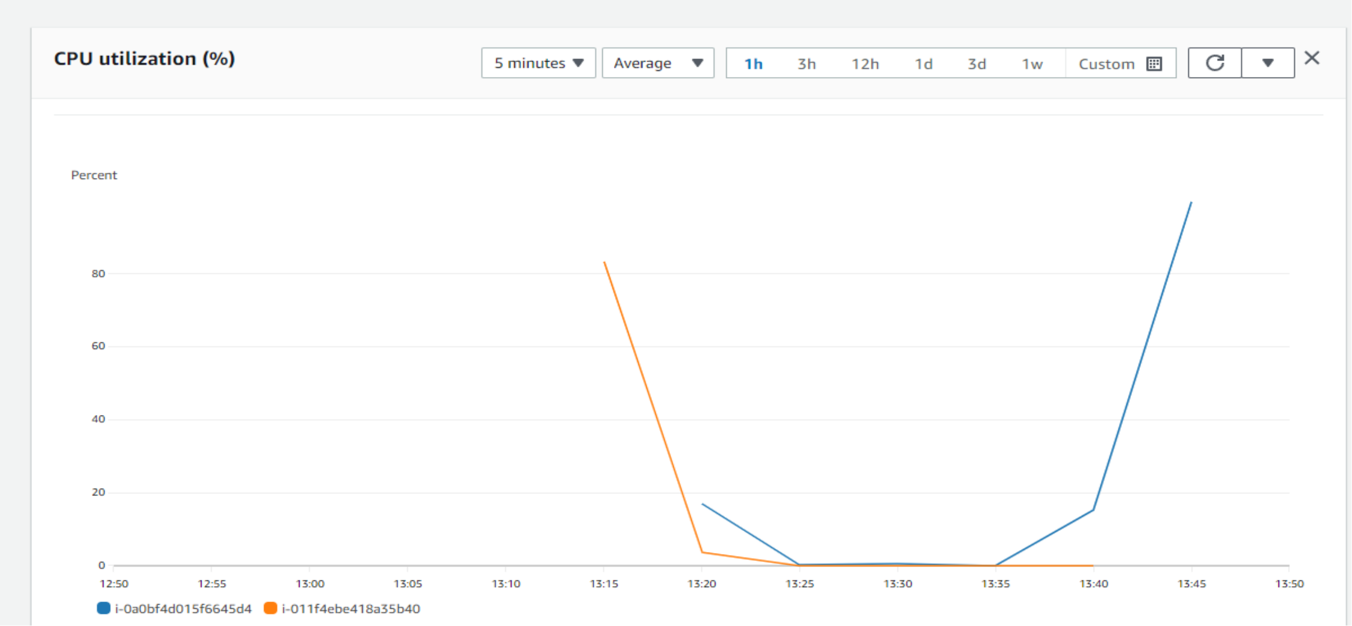
1. Click on connect option and one terminal will open.
2. In the terminal type the command as same as previous. And run .



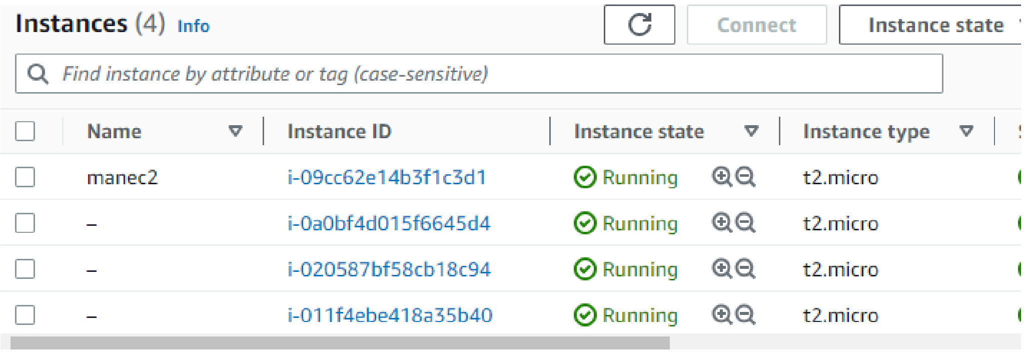
```
To run a command as administrator (user "root"),
See "man sudo_root" for details.

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.



□	Name	Instance ID	Instance state	Instance type
□	manec2	i-09cc62e14b3f1c3d1	✓ Running	t2.micro
□	–	i-0a0bf4d015f6645d4	✓ Running	t2.micro
□	–	i-020587bf58cb18c94	✓ Running	t2.micro
□	–	i-011f4ebe418a35b40	✓ Running	t2.micro