

Assignment 11: Build Scaling plans in AWS that balance load on different EC2 instances.

1. Creating Launch Template

- In the left panel, under instances, go to **Launch Templates** option and click on **Create launch template**.
- Give name and description of the template and select the checkbox. Under **Application and OS Images** under **Quick Start** select **Ubuntu**, under **Instance type** select **t2.micro**, create a key pair.

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*

Template2

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

Template version description

ver1

Max 255 chars

- Under **Network Settings**, select the **Security Group** created before.

- Go to **Advanced details** and in user data write the required data/commands.

- #!/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/Sayan-K-Dutta/Assign8-Repo2.git
- cd Assign8-Repo2
- npm install
- node index.js

(Before creating the Template make the required GitHub repo public).

- Create the template.

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 specific traffic to reach your instance.

☒ Select existing security group ☐ Create security group

Security groups Info

Select security groups

SG1 sg-0750bfbac6c367e40 X

VPC: vpc-050077ab690dd0747

[Compare security group rules](#)

[Advanced network configuration](#)

Launch templates (1/2) Info

[Refresh](#) [Actions](#) [Create launch template](#)

Filter by tags or properties or search by keyword

Launch template ID	Launch template name
<input checked="" type="radio"/> lt-0d437a6d21e853206	Template2
<input type="radio"/> lt-067405657d7886446	template1

Template2 (lt-0d437a6d21e853206)

[Actions](#) [Delete template](#)

Launch template details

Launch template ID	Launch template name
lt-0d437a6d21e853206	Template2
Default version	Owner
1	arn:aws:iam::121667217816:root

2. Creating Auto Scaling Group

- In the left panel, under **Auto Scaling**, go to **Auto Scaling Groups**. Click on **Create Auto Scaling group**.
- Give **Name** of the auto scaling group, select the launch template created and select version as **Latest(1)**. Click on **Next**.

Name

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

AS1

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

Launch template [Info](#)

Launch template
Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image and security groups.

Template2

[Create a launch template](#)

Version

Latest (1)

[Create a launch template version](#)

- Choose all the **Availability zones and subnets** and go to next page.

Availability Zones and subnets
Define which Availability Zones and subnets your Auto Scaling group can use in the cl

Select Availability Zones and subnets

ap-southeast-1c | subnet-035d4b6e609bcf043
172.31.0.0/20 Default

ap-southeast-1b | subnet-0e5e670d726b6e73f
172.31.16.0/20 Default

ap-southeast-1a | subnet-0304705e69b4b3a25
172.31.32.0/20 Default

[Create a subnet](#)

- Select **Attach to a new load balancer**. Choose Load balancer scheme as **Internet-facing**.

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

- In **Listeners and routing** give **Port** as **4000** and for **Default routing** select **Create a target group** and click the target group showing. Click on **Next**.

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): Create a target group ▼

New target group name: AS1-1
An instance target group with default settings will be created.

- f) In Group size give the **Desired**, **Minimum** and **Maximum** capacity as **2,2** and **3** respectively.
- g) In **Scaling policies** select **Target tracking scaling policy**. Select **Target value** as **50** and set Instances need **300** seconds warmup.

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: Target Tracking Policy

Metric type: Average CPU utilization ▼

Target value: 50

Instances need: 300 seconds warm up before including in metric

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

Group size - optional [Info](#)

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

Desired capacity: 2

Minimum capacity: 2

Maximum capacity: 3

- h) Go clicking on **Next** and create auto scaling group. Auto scaling group created.

EC2 > Auto Scaling groups

Auto Scaling groups (1/1) [Info](#) [Refresh](#) [Edit](#) [Delete](#) [Create an Auto Scaling group](#)

Search your Auto Scaling groups

<input checked="" type="checkbox"/>	Name	Launch template/configuration	Instances	Status	Desired capacity
<input checked="" type="checkbox"/>	AS1	Template2 Version Latest	0	Updating capacity...	2

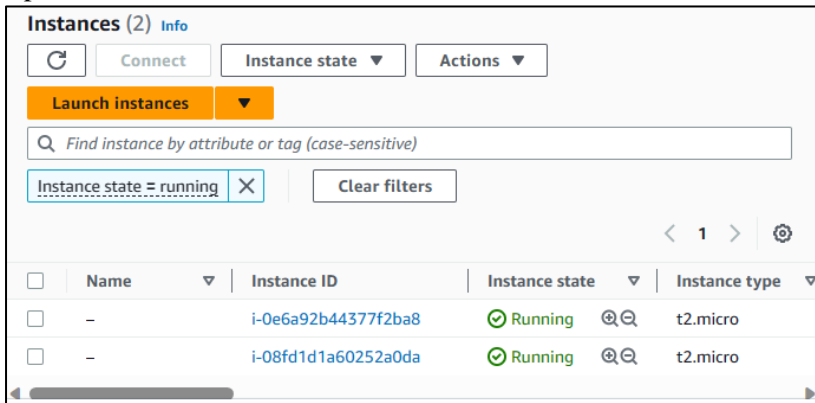
Auto Scaling group: AS1 [Settings](#) [Close](#)

[Details](#) [Activity](#) [Automatic scaling](#) [Instance management](#) [Monitoring](#) [Instance refresh](#)

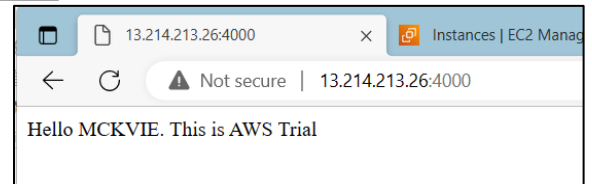
Group details [Edit](#)

Auto Scaling group name AS1	Desired capacity 2	Status Updating capacity	Amazon Resource Name (ARN) arn:aws:autoscaling:ap-southeast-1:121667217816:autoScalingGroup:p:21d02fba-ec4b-423a-88a9-e382cdad1c7a:autoScalingGroupName/AS1
Date created Fri Apr 21 2023 13:13:43 GMT+0530 (India Standard Time)	Minimum capacity 2	Maximum capacity 3	

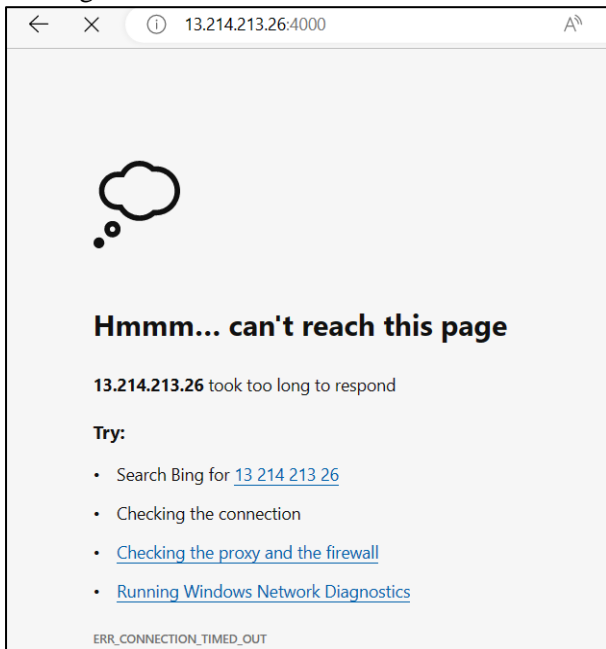
3. Open the Instances and we will see two instances created.



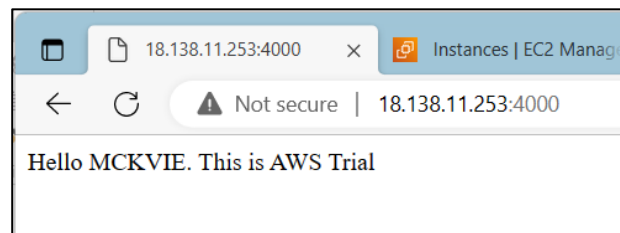
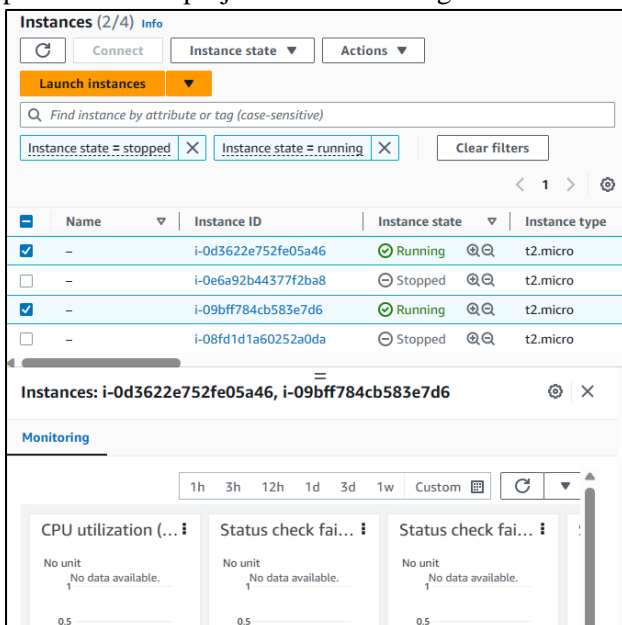
4. Copy the public IPv4 of an instance and open in web browser to see if it is working properly. Give port as 4000 to see the project running.



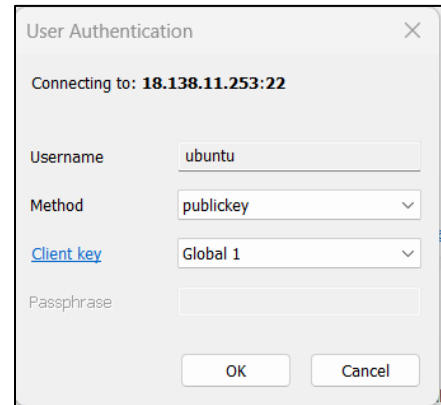
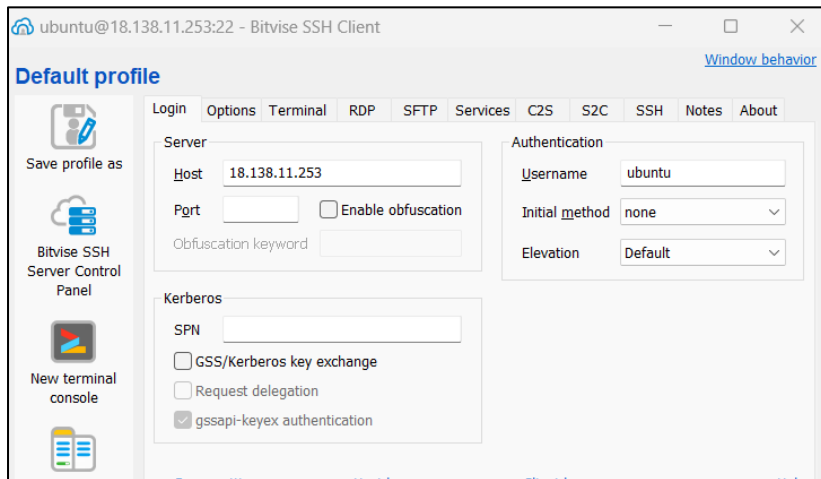
5. Now select both the instances and stop them (crashing the server) and refresh the website. We will no more see the project running.



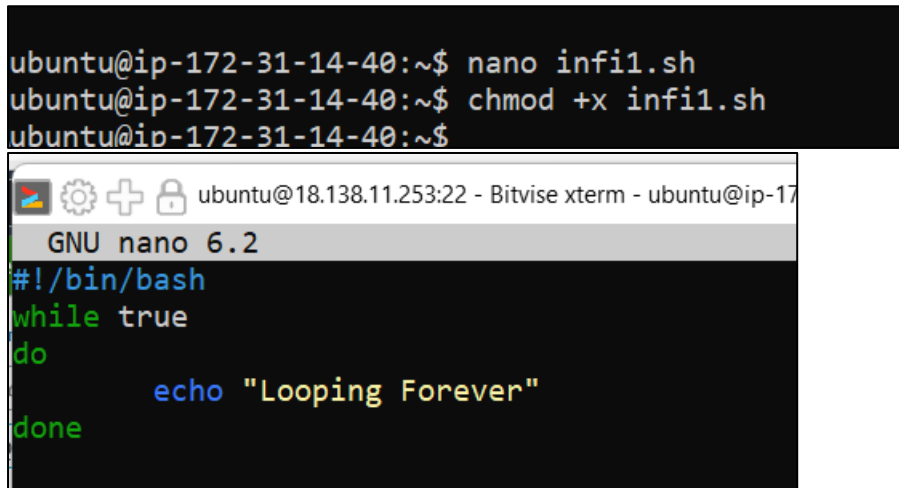
6. New instance will be created automatically. Open the project with the help of this instances' public IP address and port 4000. The project will be running.



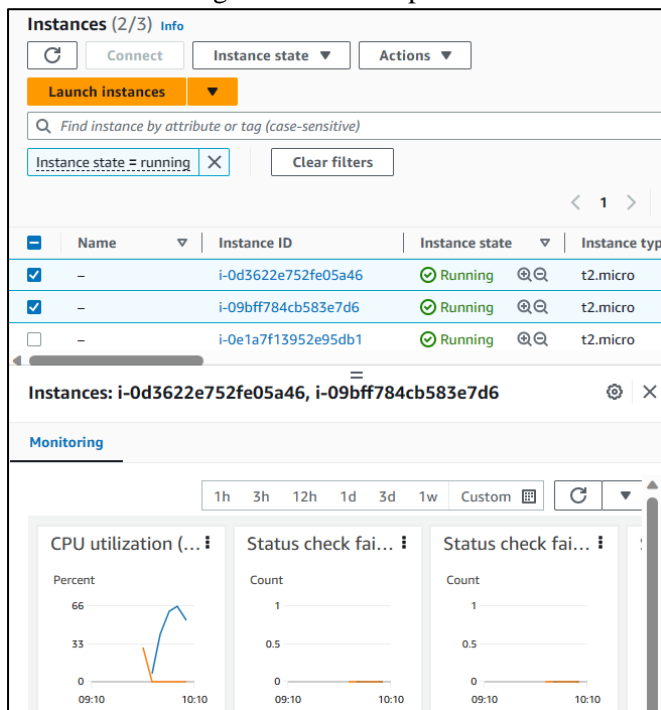
7. Open **Bitwise SSH Client** and connect a running instance.



8. Login and open a new terminal console. Write an infinite loop code save and give necessary permission (**chmod +x infi1.sh**) and execute it.



9. The server running the infinite loop will be overloaded and new instance will be created.



10. Selecting the 3 running instances, go to **Monitoring** section and **enlarge** the **CPU Utilization** section. Go to **Custom** and select the **local time zone**. We can see new instance is created when one of the servers crossed 50% and the performance of these instances.

