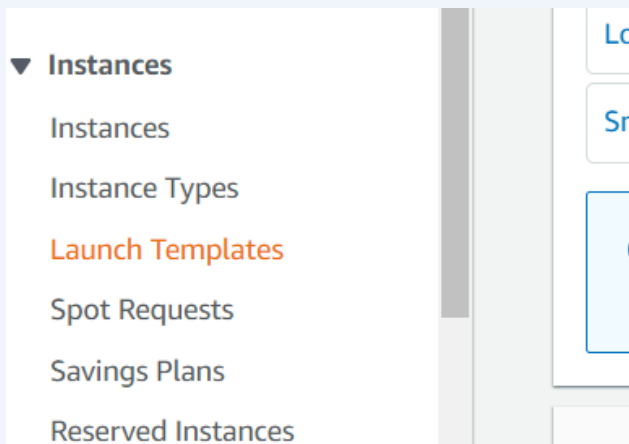


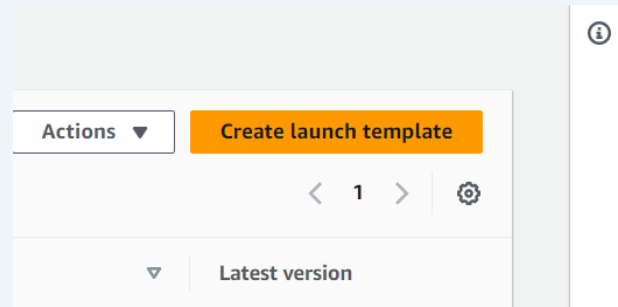
# Assignment no. 11

## Build scaling plans in AWS that balances load on different EC2 instances

1. Visit [aws.amazon.com](https://aws.amazon.com) and Sign in. Go to EC2 dashboard and click "Launch Template"



2. Click on "Create launch instance" option



3. Give template name, template version description and select autoscaling guidance

Launch template name and description

Launch template name - required

MyTemplate

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

Template version description

ver1

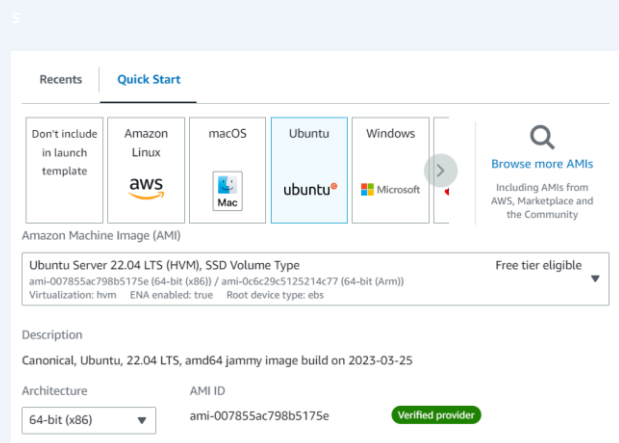
Max 255 chars

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

4. Select Ubuntu as AMI



## 5. Select “t2.micro” as instance type and key pair for login

▼ Instance type [Info](#) Advanced

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

☒ 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

key55 [Create new key pair](#)

## 6. Now select the existing security group which you created for the project

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

secure sg-0a8f06d823f059146 [X](#)

VPC: vpc-0b2db5af3ebc29dce [Compare security group rules](#)

► Advanced network configuration

## 7. Type the following user data and click on create launch template

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/hiteshperwal/hello1.git
cd hello1/
npm install
node index.js
```

**Free tier:** In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel [Create launch template](#)

## 8. In the EC2 dashboard click on “Auto scaling groups and then click on create Auto Scaling groups”

EC2 CONSOLE

Target Groups

▼ Auto Scaling

Launch Configurations

[Auto Scaling Groups](#)

### Create Auto Scaling group

Get started with EC2 Auto Scaling by creating an Auto Scaling group.

[Create Auto Scaling group](#)

9. Enter the Auto Scaling group name. Select the template created by you and its version as latest. Click on "Next"

**Name**

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

Auto

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.

MyTemplate

[Create a launch template](#)

Version

Latest (1)

[Create a launch template version](#)

Description  
ver1

Launch template  
MyTemplate  
lt-0c30a5f5aabe1a845

Instance type  
t2.micro

10. Select all the Availability zones and subnets and click on "Next"

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

vpc-0b2db5af3ebc29dcc  
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

us-east-1a | subnet-040ff3fa6f65a9a1b  
172.31.16.0/20 Default

us-east-1b | subnet-025116bf423799fb2  
172.31.32.0/20 Default

us-east-1c | subnet-07e257cda4f004993  
172.31.0.0/20 Default

us-east-1d | subnet-0c2a22a6e4867e89b  
172.31.80.0/20 Default

us-east-1e | subnet-0125834c110fd13d3  
172.31.48.0/20 Default

us-east-1f | subnet-06285a888678b6059  
172.31.64.0/20 Default

[Create a subnet](#)

11. In the load balancing ,select "Attach to a new load balancer".In the load balancer type select "Application load balancer".Give the load balancer name and select the 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 type**  
Choose from the load balancer types offered below. Type selection cannot be changed after the load balancer is created. If you need a different type of load balancer than those offered here, visit the [Load Balancing console](#).

☒ Application Load Balancer  
HTTP, HTTPS

☐ Network Load Balancer  
TCP, UDP, TLS

**Load balancer name**  
Name cannot be changed after the load balancer is created.

AutoScaling-1

**Load balancer scheme**  
Scheme cannot be changed after the load balancer is created.

☐ Internal

☒ Internet-facing

**Network mapping**  
Your new load balancer will be created using the same VPC and Availability Zone selections as your Auto Scaling group. You can select different subnets and add subnets from additional Availability Zones.

12. In the listeners section, enter the port as mentioned in your project and health check grace period as

**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  
An instance target group with default settings will be created.  
AutoScaling-1

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

[Add tag](#)  
50 remaining

**Health check grace period** [info](#)  
This time period delays the first health check until your instances finish initializing. It doesn't prevent an instance from terminating when placed into a non-running state.

300 seconds

- 13. Enter the desired capacity as 2, minimum capacity as 2 and maximum capacity as 3. Select target tracking scaling policy, give it a name, target value 50 and 300 seconds as warm up. Click on "Next", "Next", "Next" and then "Create auto scaling group".**

Group size - optional [Info](#)

Specify the size of the Auto Scaling group's maximum capacity limits. Your desired capacity must be less than or equal to the maximum capacity.

Desired capacity

Minimum capacity

Maximum capacity

Scaling policies - optional

Choose whether to use a scaling policy to dynamically resize the group in response to 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.

Scaling policy name

Metric type

Target value

Instances need

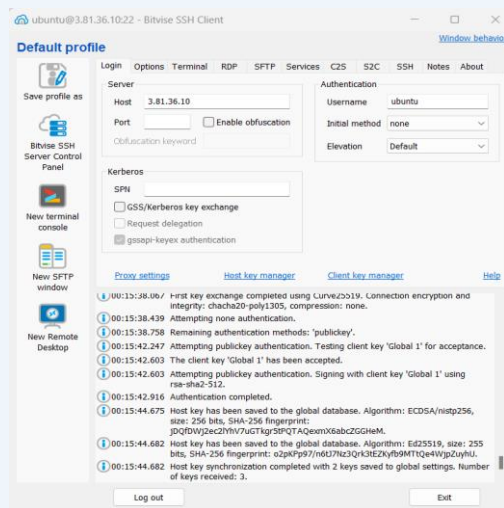
seconds warm up before including in metric

- 14.** Now two instances are running. Overload one of them to see whether auto scaling functionality is working or not.

Instances (2) Info

<input type="checkbox"/>	Name ▾	Instance ID	Instance state ▾	Instance type
<input type="checkbox"/>	-	i-0e62dbb433ee46ae3	<span>✔ Running</span> <span>🔍 🔍</span>	t2.micro
<input type="checkbox"/>	-	i-01fdaed5d372515b5	<span>✔ Running</span> <span>🔍 🔍</span>	t2.micro

15. Login in Bitwise SSH client using one of the public ip address of the instances created and the key pair used in those instance.



- 16. Open the terminal console, create a shell executables file, type the following commands and execute the file to overload the server**

```
ubuntu@ip-172-31-68-249:~$ vi inf1.sh
```

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

```
ubuntu@ip-172-31-68-249:~$ chmod +x inf1.sh
ubuntu@ip-172-31-68-249:~$ ./inf1.sh
```

[illegible]

17. Select the running instances and click on monitoring. You will observe creation of new instance during the 300 seconds warm up when the server is overloaded. Hence load is balanced on different EC2 instances

