

Assignment – 11

Problem Statement: -

Build scaling plans in AWS that balance load on different EC2 instances.

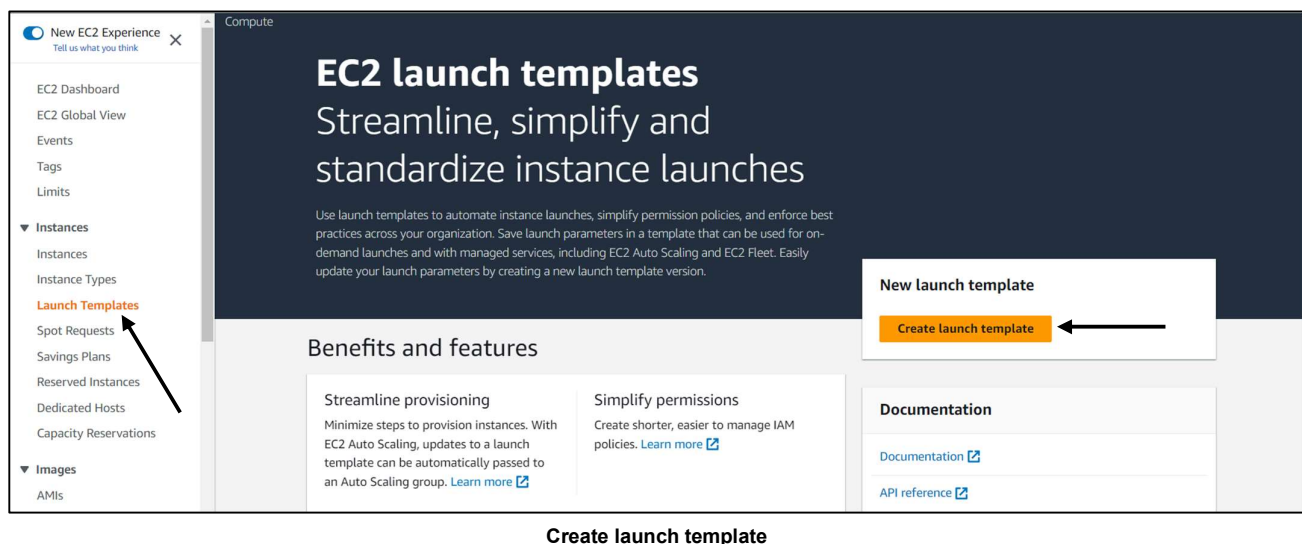
Steps for build scaling plans in AWS that balance load on different EC2 instances: -

1. First create EC2 security group with custom TCP port number 4000.
2. Go **Launch Templates** and click on **Create launch template**.
3. Fill the Launch template name and description. Select OS, Instance type, Key pair, existing security group. Also add User data –

```
#!/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/ABgithub-17/DemoProject2.git
cd DemoProject2
npm install
node index.js
```

(GitHub repository must be public)
Now, click on **Create launch template**.
4. Go back to EC2 dashboard and click on **Auto Scaling Groups**. Then click on **Create Auto Scaling group**.
5. Now, give an auto scaling group name, select newly created Launch template, choose Latest (1) as a Version of template and continue with Next.
6. In Network info, select **all Availability Zones and subnets**.
7. Choose **Attach to a new load balancer** in Load balancing info. In Attach to a new load balancer, Load balancer type **Application Load Balancer**, Load balance scheme is **Internet-facing**. Also modify **HTTP protocol port to 4000**.
8. In Group size, set **Desired capacity 2, Minimum capacity 2, Maximum capacity 3**.
9. In Scaling policies, choose **target tracking scaling policy**, set **Target value 50**.
10. Now create Auto Scaling group.
11. Go EC2 Instances, we can see 2 instances create and running. If stop the instances then automatically create new 2 instances. If we run infinite loop in both new instance and check CPU utilization. We can see instance utilize CPU approx. 99% then create another instance. Now all 3-instance run together.

Some snapshots of above process: -



EC2 > Launch templates > Create launch template

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*

TemplateAB

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

Template tags

Source template

▼ Summary

Software Image (AMI)

Canonical, Ubuntu, 22.04 LTS, ...[read more](#)

ami-02eb7a4783e7e9317

Virtual server type (instance type)

t2.micro

Firewall (security group)

securityAB

Storage (volumes)

1 volume(s) - 8 GiB

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 IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel

Create launch template

▼ 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

Q |

☐ Specify a custom value...

default

VPC: vpc-029d3ec53a57486a4

sg-0d0b3ec33ecaea884

☒ securityAB

VPC: vpc-029d3ec53a57486a4

sg-0ebc6263f75b7d222

[Compare security group rules](#)

▼ Storage (volumes) [Info](#)

EBS Volumes

[Hide details](#)

▼ Summary

Software Image (AMI)

Canonical, Ubuntu, 22.04 LTS, ...[read more](#)

ami-02eb7a4783e7e9317

Virtual server type (instance type)

t2.micro

Firewall (security group)

securityAB

Storage (volumes)

1 volume(s) - 8 GiB

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 IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel

Create launch template

Select security group

Don't include in launch template

Metadata response hop limit [Info](#)

Don't include in launch template

Allow tags in metadata [Info](#)

Don't include in 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/ABgithub-17/DemoProject2.git
cd DemoProject2
npm install
node index.js
```

☐ User data has already been base64 encoded

▼ Summary

Software Image (AMI)

Canonical, Ubuntu, 22.04 LTS, ...[read more](#)

ami-02eb7a4783e7e9317

Virtual server type (instance type)

t2.micro

Firewall (security group)

securityAB

Storage (volumes)

1 volume(s) - 8 GiB

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 IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel

Create launch template

Add User data

Launch templates (1) Info			
<input type="text" value="Filter by tags or properties or search by keyword"/>			
Launch template ID	Launch template name	Default version	Latest version
lt-00f6ec94a60ceef87	TemplateAB	1	1

Launch template created

Images

AMIs

AMI Catalog

Elastic Block Store

Volumes

Snapshots

Lifecycle Manager

Network & Security

Security Groups

Elastic IPs

Placement Groups

Key Pairs

Network Interfaces

Load Balancing

Load Balancers

Target Groups

Auto Scaling

Launch Configurations

Auto Scaling Groups

Amazon EC2 Auto Scaling

helps maintain the availability of your applications

Auto Scaling groups are collections of Amazon EC2 instances that enable automatic scaling and fleet management features. These features help you maintain the health and availability of your applications.

How it works

Pricing

Amazon EC2 Auto Scaling features have no additional fees beyond the service fees for Amazon EC2, CloudWatch (for scaling policies), and the other AWS resources that you use. Visit the pricing page of each service to learn more.

Getting started

Create Auto Scaling group

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

[Create Auto Scaling group](#)

Create Auto Scaling Group

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.

TemplateAB

↕

[Create a launch template](#)

Version

Latest (1)

↕

[Create a launch template version](#)

Description

version1

Launch template

[TemplateAB](#)
 lt-00f6ec94a60ceef87

Instance type

t2.micro

AMI ID

ami-02eb7a4783e7e9317

Security groups

-

Request Spot Instances

No

Key pair name

keyAB

Security group IDs

[sg-0ebc6263f75b7d222](#)

Additional details

Storage (volumes)

-

Date created

Wed Apr 26 2023 01:21:40
 GMT+0530 (India Standard Time)

Select launch template & Version

EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1

Choose launch template or configuration

Step 2

Choose instance launch options

Step 3 - optional

Configure advanced options

Step 4 - optional

Configure group size and scaling policies

Step 5 - optional

Add notifications

Step 6 - optional

Add tags

Step 7

Review

Choose instance launch options

Choose the VPC network environment that your instances are launched into, and customize the instance types and purchase options.

Network

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.

☒ ap-south-1a | subnet-0564cbff9731396bd

172.31.32.0/20

Default

☒ ap-south-1b | subnet-0b803b78d02cd24f1

172.31.0.0/20

Default

☒ ap-south-1c | subnet-0f56d6c967d16070f

172.31.16.0/20

Default

Select Availability Zones and subnets

ap-south-1a | subnet-0564cbff9731396bd

172.31.32.0/20

Default

ap-south-1b | subnet-0b803b78d02cd24f1

172.31.0.0/20

Default

ap-south-1c | subnet-0f56d6c967d16070f

172.31.16.0/20

Default

Select Availability Zones & subnets

EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1

Choose launch template or configuration

Step 2

Choose instance launch options

Step 3 - optional

Configure advanced options

Step 4 - optional

Configure group size and scaling policies

Step 5 - optional

Add notifications

Step 6 - optional

Add tags

Step 7

Review

Configure advanced options - optional

Choose a load balancer to distribute incoming traffic for your application across instances to make it more reliable and easily scalable. You can also set options that give you more control over health check replacements and monitoring.

Load balancing

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.

Attach to a new load balancer

Define a new load balancer to create for attachment to this 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

Configure Load balancer

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.

VPC

vpc-029d3ec53a57486a4

Availability Zones and subnets

You must select a single subnet for each Availability Zone enabled. Only public subnets are available for selection to support DNS resolution.

☒ ap-south-1b

subnet-0b803b78d02cd24f1

☒ ap-south-1a

subnet-0564cbff9731396bd

☒ ap-south-1c

subnet-0f56d6c967d16070f

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

autoscalingAB-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.

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

Scaling policies

Auto Scaling groups (1) [Info](#)

Search your Auto Scaling groups

1

<input type="checkbox"/>	Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availabil...
<input type="checkbox"/>	autoscalingAB	TemplateAB Version Latest	0	Updating capacity...	2	2	3	ap-south-1...

Auto Scaling Group created

New EC2 Experience

EC2 Dashboard

EC2 Global View

Events

Tags

Limits

Instances

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity Reservations

Images

AMIs

AMI Catalog

Elastic Block Store

Volumes

Instances (2/2) [Info](#)

Find instance by attribute or tag (case-sensitive)

Instance state = running

Clear filters

<input checked="" type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status	Availability Zone	Public IPv4 DNS
<input checked="" type="checkbox"/>	-	i-03a8900296dcc27d2	Running	t2.micro	2/2	ap-south-1b	ec2-15-206-84-9
<input checked="" type="checkbox"/>	-	i-0adfae15d35322041	Running	t2.micro	2/2 checks passed	ap-south-1a	ec2-13-235-244-

Instances: i-03a8900296dcc27d2, i-0adfae15d35322041

Monitoring

1h 3h 12h 1d 3d 1w Custom

CPU utilization (%)

Status check failed (any) (cou...)

Status check failed (instance)...

Status check failed (system) (...)

Two EC2 Instance create

New EC2 Experience

Tell us what you think

EC2 Dashboard

EC2 Global View

Events

Tags

Limits

Instances (5) Info

Find instance by attribute or tag (case-sensitive)

1

Connect

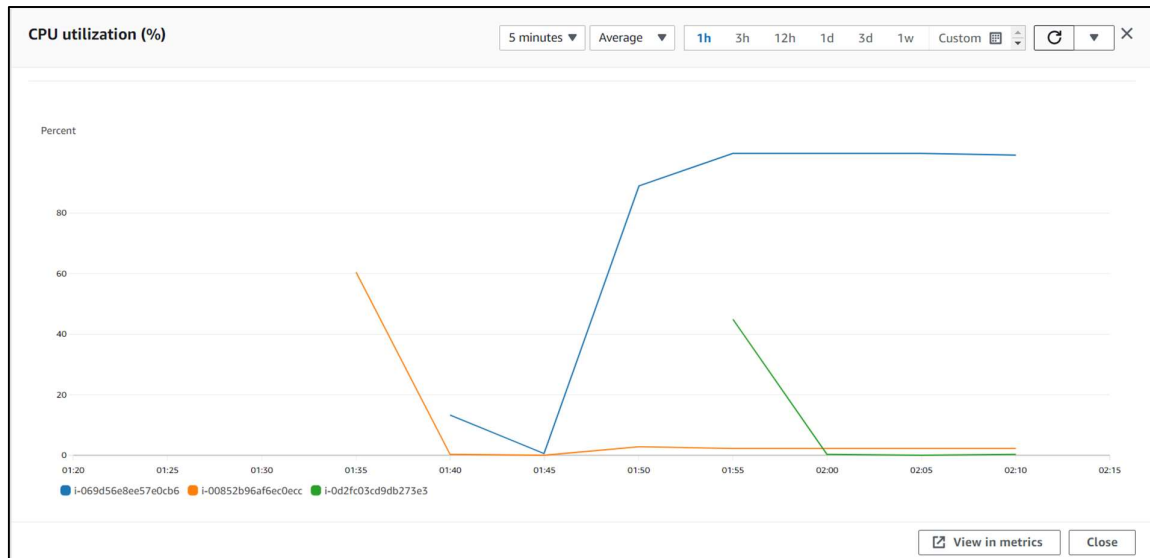
Instance state

Actions

Launch instances

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
<input type="checkbox"/>	-	i-03a8900296dcc27d2	Terminated	t2.micro	-	No alarms	ap-south-1b	-
<input type="checkbox"/>	-	i-069d56e8ee57e0cb6	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1b	ec2-13-233-54-4
<input type="checkbox"/>	-	i-0adfae15d35322041	Terminated	t2.micro	-	No alarms	ap-south-1a	-
<input type="checkbox"/>	-	i-00852b96af6ec0ecc	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a	ec2-52-66-202-2

Two new instances created after stopping previous two instance



New EC2 Experience

Tell us what you think

EC2 Dashboard

EC2 Global View

Events

Tags

Limits

Instances

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity Reservations

Images

AMIs

AMI Catalog

Elastic Block Store

Volumes

Instances (3/3) Info

Find instance by attribute or tag (case-sensitive)

Instance state = running

Clear filters

Connect

Instance state

Actions

Launch instances

<input checked="" type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
<input checked="" type="checkbox"/>	-	i-069d56e8ee57e0cb6	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1b	ec2-13-233-54-4
<input checked="" type="checkbox"/>	-	i-00852b96af6ec0ecc	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a	ec2-52-66-202-2
<input checked="" type="checkbox"/>	-	i-0d2fc03cd9db273e3	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a	ec2-65-0-92-171

Instances: i-069d56e8ee57e0cb6, i-00852b96af6ec0ecc, i-0d2fc03cd9db273e3

Monitoring

1h 3h 12h 1d 3d 1w Custom

CPU utilization (%)

Percent

99.7

49.9

0

19:35

20:35

Status check failed (any) (cou...)

Various units

1

0.5

0

19:35

20:35

Status check failed (instance)...

Various units

1

0.5

0

19:35

20:35

Status check failed (system) (...)

Various units

1

0.5

0

19:35

20:35

Add to dashboard

Create 3rd instance