

## AWS AUTO SCALING

*AWS Auto Scaling is a service that automatically adjusts the number of instances in your AWS setup based on the conditions you define.*

*AWS Auto Scaling helps you maintain the availability and performance of your applications by automatically adjusting the number of EC2 instances or other resources based on demand. When demand increases, it adds more instances; when demand decreases, it removes instances, ensuring you're not paying for resources you don't need.*

### Types of Auto Scaling:

1. **Manual Scaling:** In the AWS Auto Scaling console, you can create scaling policies that specify the conditions under which scaling actions should be taken. These policies can be triggered based on metrics like CPU utilization, network traffic, or custom metrics that you define. By configuring these policies manually, you effectively implement manual scaling.
2. **Dynamic Scaling:** When creating scaling policies, you can choose to use dynamic scaling by configuring scaling actions based on CloudWatch alarms or other metric data. These policies will automatically adjust the number of instances based on real-time demand, effectively implementing dynamic scaling.
3. **Scheduled Scaling:** In the AWS Auto Scaling console, you can create scheduled actions that define specific times for scaling actions to occur. These actions allow you to anticipate changes in demand and adjust your resources accordingly, implementing scheduled scaling.
4. **Predictive Scaling:** While not explicitly labeled in the console, you can leverage AWS Auto Scaling features like predictive scaling by enabling predictive scaling within your scaling policies. AWS Auto Scaling uses machine learning algorithms to forecast future demand based on historical data and adjusts resources proactively, optimizing performance and cost.

**Note:** while the AWS Management Console may not have distinct options labeled as "Auto Scaling Types," you can implement different types of auto scaling by configuring your scaling policies and settings according to your specific requirements and preferences.

---

*Vertical and horizontal scaling are indeed approaches to scaling infrastructure but are not typically referred to as types of auto scaling within AWS.*

1. **Vertical Auto Scaling:**
  - Also known as scaling up or scaling vertically.
  - Involves increasing the capacity of individual instances by adding more resources to them, such as CPU, memory, or storage.
  - Example: Instead of adding more instances to your fleet, you might vertically scale by upgrading the instance type to a larger one with more CPU and memory.
2. **Horizontal Auto Scaling:**
  - Also known as scaling out or scaling horizontally.
  - Involves adding more instances to your application to distribute the load across multiple servers.
  - Example: When demand increases, horizontal scaling automatically spins up additional instances of your application to handle the increased load. Conversely, when demand decreases, it reduces the number of instances to save costs.

*Both vertical and horizontal scaling have their advantages and use cases, and in many scenarios, a combination of both is employed to ensure optimal performance and resource utilization in response to changing demands.*

Now we will set up auto scaling for an application, we can set auto scaling for our running instance or even create a new instance using a template.

We have a pre-configured Instance template, and we need to create a Load balancer, and Target group then we create an Auto Scaling group by configuring our template and load balancer to it.

Then we will put artificial stress on our servers to test whether our Auto Scaling is working properly.

[Steps to Setting up Auto Scaling](#)

- Step 1) Go to **Ec2 Dashboard's** Auto Scaling Menu, select **Auto Scaling Groups**,  
Step 2) Click on **Create Auto Scaling Group**, enter the name of your ASG

### Choose launch template or configuration Info

Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group. If you currently use launch configurations, you might consider migrating to launch templates.

**Name**

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

Test-ASG-for-Ec2

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

- Step 3) Select Template, and click on the next button. (if the pre-configured template is not created click on Create a Launch Template, and a new tab will open Launch it then proceed next step)

### Choose launch template or configuration Info

Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group. If you currently use launch configurations, you might consider migrating to launch templates.

**Name**

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

Ec2-Test-ASG

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

**Launch template** Info [Switch to launch configuration](#)

TestASG.Temp

TestASG.Temp

Create a launch template

Version

Default (1)

Create a launch template version

- Step 4) In the Instance type requirement, you can either use the same instance settings as your launch template or specify different settings manually. Now we go with the template setting.

### Instance type requirements Info

Override launch template

You can keep the same instance attributes or instance type from your launch template, or you can choose to override the launch template by specifying different instance attributes or manually adding instance types.

Launch template	Version	Description
TestASG.Temp	Default	ASG
lt-0d42191ee4253731c		
Instance type		
t2.micro		

Step 5) In the Network option, we select VPC and define the availability zone and subnets, where our instance will launch. Then click on the next button.

NetworkInfo

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.

vpc-0b61408ca59d3f247

172.31.0.0/16Default

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-05722504926d6dec4

172.31.16.0/20Default

☒us-east-1b | subnet-0e4e8a76d237d9dae

172.31.32.0/20Default

☒us-east-1c | subnet-04496339d5f219a6f

172.31.0.0/20Default

☒us-east-1d | subnet-008527ce3ecaebe76

172.31.80.0/20Default

☒us-east-1e | subnet-0a7fcbaa05abb5f51

172.31.48.0/20Default

☒us-east-1f | subnet-0d269a381e3d7b4f3

172.31.64.0/20Default

Step 6) In the Load balancing option, attach to an existing load balancer. Select load balancer and target group.

Load balancingInfo

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 an existing load balancer

Select the load balancers that you want to attach to your Auto Scaling group.

☒Choose from your load balancer target groups

This option allows you to attach Application, Network, or Gateway Load Balancers.

☐Choose from Classic Load Balancers

Existing load balancer target groups

Only instance target groups that belong to the same VPC as your Auto Scaling group are available for selection.

Select target groups

TargetASG | HTTP

Application Load Balancer: ASG-ALB

Also, to have more options as shown in the image we can create a new load balancer, and in this have the option to create target groups.

Other settings rest as by default.

*Step 7) Now this is our important step here we configure group size and scaling. Shown as in the image.*

### Configure group size and scaling - optional [Info](#)

Define your group's desired capacity and scaling limits. You can optionally add automatic scaling to adjust the size of your group.

#### Group size [Info](#)

Set the initial size of the Auto Scaling group. After creating the group, you can change its size to meet demand, either manually or by using automatic scaling.

#### Desired capacity type

Choose the unit of measurement for the desired capacity value. vCPUs and Memory(GiB) are only supported for mixed instances groups configured with a set of instance attributes.

Units (number of instances) ▼

#### Desired capacity

Specify your group size.

4 ←

### Scaling [Info](#)

You can resize your Auto Scaling group manually or automatically to meet changes in demand.

#### Scaling limits

Set limits on how much your desired capacity can be increased or decreased.

##### Min desired capacity

2 ←

Equal or less than desired capacity

##### Max desired capacity

8 ←

Equal or greater than desired capacity

#### Automatic scaling - optional

Choose whether to use a target tracking policy [Info](#)

You can set up other metric-based scaling policies and scheduled scaling after creating your Auto Scaling group.

☐ No scaling policies

Your Auto Scaling group will remain at its initial size and will not dynamically resize to meet demand.

☒ Target tracking scaling policy

Choose a CloudWatch metric and target value and let the scaling policy adjust the desired capacity in proportion to the metric's value.

#### Scaling policy name

Target Tracking Policy

#### Metric type [Info](#) ←

Monitored metric that determines if resource utilization is too low or high. If using EC2 metrics, consider enabling detailed monitoring for better scaling performance.

Average CPU utilization ▼ ←

#### Target value

50 ←

#### Instance warmup [Info](#)

60 seconds ←

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

**Define desired capacity:** Set the number of instances you want to maintain in your Auto Scaling group.

**Create scaling policies:** Configure rules to automatically adjust instance count based on metrics like CPU usage or network traffic.

**Set minimum and maximum instances:** Specify the range of instances allowed to ensure scalability and cost control.

**Set cooldown periods:** Add cooldown periods to prevent rapid scaling actions in response to fluctuations.

Step 8) Select Instance maintenance policy. click on the next button

Instance maintenance policy - new Info

Control your Auto Scaling group's availability during instance replacement events. This includes health checks, instance refreshes, maximum instance lifetime features and events that happen automatically to keep your group balanced, called rebalancing events.

Control availability and cost during replacement events

An instance maintenance policy determines how much availability your application has when EC2 Auto Scaling replaces instances. It also establishes guardrails that limit the amount of capacity that can be added or removed when replacing instances.

Choose a replacement behavior depending on your availability requirements

Mixed behavior

No policy

For rebalancing events, new instances will launch before terminating others. For all other events, instances terminate and launch at the same time.

Prioritize availability

Launch before terminating

Launch new instances and wait for them to be ready before terminating others. This allows you to go above your desired capacity by a given percentage and may temporarily increase costs.

Control costs

Terminate and launch

Terminate and launch instances at the same time. This allows you to go below your desired capacity by a given percentage and may temporarily reduce availability.

Flexible

Custom behavior

Set custom values for the minimum and maximum amount of available capacity. This gives you greater flexibility in setting how far below and over your desired capacity EC2 Auto Scaling goes when replacing instances.

The instance maintenance policy defines how instances should be managed during scheduled events or when a problem occurs. These policies ensure that your applications remain available and responsive even when instances are being replaced or taken out of service.

Step 9) Add notifications option, is optional, click on the next button.

Add notifications - optional Info

Send notifications to SNS topics whenever Amazon EC2 Auto Scaling launches or terminates the EC2 instances in your Auto Scaling group.

Add notification

Cancel

Skip to review

Previous

Next

Step 10) Add Tag option, Add or you can skip by clicking the Next button.

Step 11) Review page: check all details, Configurations, and settings are correct according to your use case, then simply click on Create Auto Scaling Group Button.

Now you will see your Auto Scaling Group is created. It will take some time to work, go to the activity, and check the activity history. Our instance will launched successfully.

Activity history (4)					
Filter activity history					
Status	Description	Cause	Start time	End time	
Successful	Launching a new EC2 instance: i-0a58be6ee73500f92	At 2024-03-05T17:12:34Z a user request created an AutoScalingGroup changing the desired capacity from 0 to 4. At 2024-03-05T17:12:46Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 4.	2024 March 05, 10:42:50 PM +05:30	2024 March 05, 10:43:22 PM +05:30	
Successful	Launching a new EC2 instance: i-0d0815198dd33feeb	At 2024-03-05T17:12:34Z a user request created an AutoScalingGroup changing the desired capacity from 0 to 4. At 2024-03-05T17:12:46Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 4.	2024 March 05, 10:42:50 PM +05:30	2024 March 05, 10:43:22 PM +05:30	
Successful	Launching a new EC2 instance: i-00cb87c4610e94157	At 2024-03-05T17:12:34Z a user request created an AutoScalingGroup changing the desired capacity from 0 to 4. At 2024-03-05T17:12:46Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 4.	2024 March 05, 10:42:50 PM +05:30	2024 March 05, 10:43:22 PM +05:30	
Successful	Launching a new EC2 instance: i-0148d792cd739223d	At 2024-03-05T17:12:34Z a user request created an AutoScalingGroup changing the desired capacity from 0 to 4. At 2024-03-05T17:12:46Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 4.	2024 March 05, 10:42:48 PM +05:30	2024 March 05, 10:43:20 PM +05:30	

Step 12) Go to the Ec2 Instance dashboard and check whether Instances are properly running or not. We will see our 4 instances are properly running

Instances (4) Info

Find Instance by attribute or tag (case-sensitive)

Any state

Refresh

Connect

Instance state

Actions

Launch instances

	Name	Instance ID	Instance state	Instanc...	Status check	Alarm status	Availability Zo...	Public IPv4 DNS	Public IPv4 ...	Elastic IP
<input type="checkbox"/>	1	i-0a58be6ee7350...	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1e	ec2-54-227-128-72.co...	54.227.128.72	-
<input type="checkbox"/>	2	i-0d0815198dd33...	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1d	ec2-35-175-149-63.co...	35.175.149.63	-
<input type="checkbox"/>	3	i-00cb87c4610e9...	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1c	ec2-44-201-42-89.com...	44.201.42.89	-
<input type="checkbox"/>	4	i-0148d792cd739...	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1f	ec2-3-235-242-39.com...	3.235.242.39	-

Step 13) Now do nothing for 15 to 20 min. and then check the activity history in the auto scaling group.

Successful	Terminating EC2 instance: i-00cb87c4610e94157	At 2024-03-05T17:29:07Z a monitor alarm TargetTracking-Ec2-Test-ASG-AlarmLow-6d66bbf5-48d4-4797-99d2-1034bca9b094 in state ALARM triggered policy Target Tracking Policy changing the desired capacity from 3 to 2. At 2024-03-05T17:29:15Z an instance was taken out of service in response to a difference between desired and actual capacity, shrinking the capacity from 3 to 2. At 2024-03-05T17:29:15Z instance i-00cb87c4610e94157 was selected for termination.	2024 March 05, 10:59:15 PM +05:30	2024 March 05, 11:05:08 PM +05:30
Successful	Terminating EC2 instance: i-0148d792cd739223d	At 2024-03-05T17:28:48Z a monitor alarm TargetTracking-Ec2-Test-ASG-AlarmLow-339f65fc-0f85-436a-b5c9-c48b95aecc6d in state ALARM triggered policy Target Tracking Policy changing the desired capacity from 4 to 3. At 2024-03-05T17:29:04Z an instance was taken out of service in response to a difference between desired and actual capacity, shrinking the capacity from 4 to 3. At 2024-03-05T17:29:04Z instance i-0148d792cd739223d was selected for termination.	2024 March 05, 10:59:04 PM +05:30	2024 March 05, 11:04:54 PM +05:30

Instances (10) Info

Find Instance by attribute or tag (case-sensitive)

Any state

Refresh

Connect

Instance state

Actions

Launch instances

	Name	Instance ID	Instance state	Instanc...	Status check	Alarm status	Availability Zo...	Public IPv4 DNS	Public IPv4 ...	Elastic IP
<input type="checkbox"/>	4	i-0148d792cd739...	Terminated	t2.micro	-	View alarms +	us-east-1f	-	-	-
<input type="checkbox"/>	3	i-00cb87c4610e9...	Terminated	t2.micro	-	View alarms +	us-east-1c	-	-	-
<input type="checkbox"/>	2	i-0d0815198dd33...	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1d	ec2-35-175-149-63.co...	35.175.149.63	-
<input type="checkbox"/>	1	i-0a58be6ee7350...	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1e	ec2-54-227-128-72.co...	54.227.128.72	-

In the above image, we can see that the 3<sup>rd</sup> and 4<sup>th</sup> Instances have terminated because we have set the minimum desired capacity at 2 when no more traffic or utilization.

Step 13) Now connect to Instances and put artificial stress into the CPU utilization.

We need to install the Stress Package

Commands to install Stress Package: -> yum install stress -y

To check all options: -> stress --help

Simply copy the example command from the help menu, Paste, and change the 80 % stress level of CPU and period for 20 min then run it.

```
root@ip-172-31-49-113:~
[root@ip-172-31-49-113 ~]# stress --cpu 80 --io 4 --vm 2 --vm-bytes 128M --timeout 20m
stress: info: [26263] dispatching hogs: 80 cpu, 4 io, 2 vm, 0 hdd
```

Do the same with the 2<sup>nd</sup> instance.

Auto Scaling Group will automatically launch new instances when the instance CPU utilization reaches 80%.



Step 14) Again go to the activity history option of auto scaling group and check the new activity.

Successful	Launching a new EC2 instance: i-07dcfc399d89f5fa9	At 2024-03-05T17:38:59Z a monitor alarm TargetTracking-Ec2-Test-ASG-AlarmHigh-511a2c40-f0d0-4a03-a509-6fe03b75ed6d in state ALARM triggered policy Target Tracking Policy changing the desired capacity from 6 to 8. At 2024-03-05T17:39:11Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 6 to 8.	2024 March 05, 11:09:13 PM +05:30	2024 March 05, 11:10:45 PM +05:30
Successful	Launching a new EC2 instance: i-0be49e767c3f2e245	At 2024-03-05T17:38:59Z a monitor alarm TargetTracking-Ec2-Test-ASG-AlarmHigh-511a2c40-f0d0-4a03-a509-6fe03b75ed6d in state ALARM triggered policy Target Tracking Policy changing the desired capacity from 6 to 8. At 2024-03-05T17:39:11Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 6 to 8.	2024 March 05, 11:09:13 PM +05:30	2024 March 05, 11:10:45 PM +05:30
Successful	Launching a new EC2 instance: i-07ec8c664ef32e606	At 2024-03-05T17:35:59Z a monitor alarm TargetTracking-Ec2-Test-ASG-AlarmHigh-511a2c40-f0d0-4a03-a509-6fe03b75ed6d in state ALARM triggered policy Target Tracking Policy changing the desired capacity from 4 to 6. At 2024-03-05T17:36:04Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 4 to 6.	2024 March 05, 11:06:07 PM +05:30	2024 March 05, 11:07:38 PM +05:30
Successful	Launching a new EC2 instance: i-0e4493345e3a31b94	At 2024-03-05T17:35:59Z a monitor alarm TargetTracking-Ec2-Test-ASG-AlarmHigh-511a2c40-f0d0-4a03-a509-6fe03b75ed6d in state ALARM triggered policy Target Tracking Policy changing the desired capacity from 4 to 6. At 2024-03-05T17:36:04Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 4 to 6.	2024 March 05, 11:06:06 PM +05:30	2024 March 05, 11:07:38 PM +05:30
Successful	Launching a new EC2 instance: i-09a6133c903e91b3a	At 2024-03-05T17:33:59Z a monitor alarm TargetTracking-Ec2-Test-ASG-AlarmHigh-511a2c40-f0d0-4a03-a509-6fe03b75ed6d in state ALARM triggered policy Target Tracking Policy changing the desired capacity from 2 to 4. At 2024-03-05T17:34:01Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 2 to 4.	2024 March 05, 11:04:03 PM +05:30	2024 March 05, 11:05:35 PM +05:30
Successful	Launching a new EC2 instance: i-09e88db157df386ce	At 2024-03-05T17:33:59Z a monitor alarm TargetTracking-Ec2-Test-ASG-AlarmHigh-511a2c40-f0d0-4a03-a509-6fe03b75ed6d in state ALARM triggered policy Target Tracking Policy changing the desired capacity from 2 to 4. At 2024-03-05T17:34:01Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 2 to 4.	2024 March 05, 11:04:03 PM +05:30	2024 March 05, 11:05:35 PM +05:30

We will see that the new 6 instances have been launched successfully.

Step 15) Go to the Ec2 Instance dashboard and check whether Instances are properly running or not. We will see our 6 instances are properly running.

Instances (10) <span>Info</span>											<span>Refresh</span>	<span>Connect</span>	<span>Instance state</span>	<span>Actions</span>	<span>Launch instances</span>
<input type="text" value="Find Instance by attribute or tag (case-sensitive)"/>											<span>Any state</span>				
<input type="checkbox"/>	Name	Instance ID	Instance state	Instanc...	Status check	Alarm status	Availability Zo...	Public IPv4 DNS	Public IPv4 ...	Elastic IP					
<input type="checkbox"/>	4	i-0148d792cd739...	Terminated	t2.micro	-	<a href="#">View alarms</a>	us-east-1f	-	-	-					
<input type="checkbox"/>	3	i-00cb87c4610e9...	Terminated	t2.micro	-	<a href="#">View alarms</a>	us-east-1c	-	-	-					
<input type="checkbox"/>	2	i-0d0815198dd33...	Running	t2.micro	2/2 checks passed	<a href="#">View alarms</a>	us-east-1d	ec2-35-175-149-63.co...	35.175.149.63	-					
<input type="checkbox"/>	1	i-0a58be6ee7350...	Running	t2.micro	2/2 checks passed	<a href="#">View alarms</a>	us-east-1e	ec2-54-227-128-72.co...	54.227.128.72	-					
<input type="checkbox"/>		i-07dcfc399d89f5...	Running	t2.micro	2/2 checks passed	<a href="#">View alarms</a>	us-east-1e	ec2-52-86-72-6.comput...	52.86.72.6	-					
<input type="checkbox"/>		i-07ec8c664ef32e...	Running	t2.micro	2/2 checks passed	<a href="#">View alarms</a>	us-east-1c	ec2-44-204-125-55.co...	44.204.125.55	-					
<input type="checkbox"/>		i-09e88db157df3...	Running	t2.micro	2/2 checks passed	<a href="#">View alarms</a>	us-east-1b	ec2-54-146-108-94.co...	54.146.108.94	-					
<input type="checkbox"/>		i-0be49e767c3f2e...	Running	t2.micro	2/2 checks passed	<a href="#">View alarms</a>	us-east-1f	ec2-44-220-53-83.com...	44.220.53.83	-					
<input type="checkbox"/>		i-0e4493345e3a3...	Running	t2.micro	2/2 checks passed	<a href="#">View alarms</a>	us-east-1f	ec2-44-197-201-80.co...	44.197.201.80	-					
<input type="checkbox"/>		i-09a6133c903e9...	Running	t2.micro	2/2 checks passed	<a href="#">View alarms</a>	us-east-1a	ec2-54-242-142-248.co...	54.242.142.248	-					

We set the maximum desired capacity at 6, so as the stress on the Instances increased, 4 more Instances were launched by Auto Scaling Group.