Amazon EC2 Auto Scaling

Neal

Please use the menu below to navigate the article sections: Hide article menu

AWS Auto Scaling monitors your applications and automatically adjusts capacity to maintain steady, predictable performance at the lowest possible cost.

AWS Auto Scaling refers to a collection of Auto Scaling capabilities across several AWS services.

The services within the AWS Auto Scaling family include:

- Amazon EC2 (known as Amazon EC2 Auto Scaling).
- Amazon ECS.
- Amazon DynamoDB.
- Amazon Aurora.

This page is specifically for Amazon EC2 Auto Scaling – Auto Scaling will also be discussed for the other services on their respective pages.

Amazon EC2 Auto Scaling Features

Amazon EC2 Auto Scaling helps you ensure that you have the correct number of Amazon EC2 instances available to handle the load for your application.

You create collections of EC2 instances, called Auto Scaling groups.

Automatically provides horizontal scaling (scale-out) for your instances.

Triggered by an event of scaling action to either launch or terminate instances.

Availability, cost, and system metrics can all factor into scaling.

Auto Scaling is a region-specific service.

Auto Scaling can span multiple AZs within the same AWS region.

Auto Scaling can be configured from the Console, CLI, SDKs and APIs.

There is no additional cost for Auto Scaling, you just pay for the resources (EC2 instances) provisioned.

Auto Scaling works with ELB, CloudWatch and CloudTrail.

You can determine which subnets Auto Scaling will launch new instances into.

Auto Scaling will try to distribute EC2 instances evenly across AZs.

Launch configuration is the template used to create new EC2 instances and includes parameters such as instance family, instance type, AMI, key pair, and security groups.

You cannot edit a launch configuration once defined.

A launch configuration:

- Can be created from the AWS console or CLI.
- You can create a new launch configuration, or.
- You can use an existing running EC2 instance to create the launch configuration.
    - The AMI must exist on EC2.

    - EC2 instance tags and any additional block store volumes created after the instance launch will not be considered.
- If you want to change your launch configurations you have to create a new one, make the required changes, and use that with your auto scaling groups.

You can use a launch configuration with multiple Auto Scaling Groups (ASG).

Launch templates are similar to launch configurations and offer more options (more below).

An Auto Scaling Group (ASG) is a logical grouping of EC2 instances managed by an Auto Scaling Policy.

An ASG can be edited once defined.

You can attach one or more classic ELBs to your existing ASG.

You can attach one or more Target Groups to your ASG to include instances behind an ALB.

The ELBs must be in the same region.

Once you do this any EC2 instance existing or added by the ASG will be automatically registered with the ASG defined ELBs.

If adding an instance to an ASG would result in exceeding the maximum capacity of the ASG the request will fail.

You can add a running instance to an ASG if the following conditions are met:

- The instance is in a running state.
- The AMI used to launch the instance still exists.
- The instance is not part of another ASG.
- The instance is in the same AZs for the ASG.

Scaling Options

The scaling options define the triggers and when instances should be provisioned/de-provisioned.

There are four scaling options:

- Maintain – keep a specific or minimum number of instances running.
- Manual – use maximum, minimum, or a specific number of instances.
- Scheduled – increase or decrease the number of instances based on a schedule.
- Dynamic – scale based on real-time system metrics (e.g. CloudWatch metrics).
- Predictive – machine learning to schedule the right number of EC2 instances in anticipation of approaching traffic changes.

The following table describes the scaling options available and when to use them:

Scaling	Description	When to use
Maintain	Ensures the required number of instances are running	Use when you always need a known number of instances running at all times
Manual	Manually change desired capacity	Use when your needs change rarely enough that you're ok the make manual changes
Scheduled	Adjust min/max on specific dates/times or recurring time periods	Use when you know when your busy and quiet times are. Useful for ensuring enough instances are available before very busy times
Dynamic	Scale in response to system load or other triggers using metrics	Useful for changing capacity based on system utilization, e.g. CPU hits 80%.

Predictive	predict capacity required ahead of time using ML	Useful for when capacity, and number of instances is unknown.

Scheduled Scaling

Scaling based on a schedule allows you to scale your application ahead of predictable load changes.

For example, you may know that traffic to your application is highest between 9am and 12pm Monday-Friday.

Dynamic Scaling

Amazon EC2 Auto Scaling enables you to follow the demand curve for your applications closely, reducing the need to manually provision Amazon EC2 capacity in advance.

For example, you can track the CPU utilization of your EC2 instances or the “Request Count Per Target” to track the number of requests coming through an Application Load Balancer.

Amazon EC2 Auto Scaling will then automatically adjust the number of EC2 instances as needed to maintain your target.

Predictive Scaling

Predictive Scaling uses machine learning to schedule the optimum number of EC2 instances in anticipation of upcoming traffic changes.

Predictive Scaling predicts future traffic, including regularly occurring spikes, and provisions the right number of EC2 instances in advance.

Predictive Scaling uses machine learning algorithms to detect changes in daily and weekly patterns and then automatically adjust forecasts.

You can configure the scaling options through Scaling Policies which determine when, if, and how the ASG scales out and in.

The following table describes the scaling policy types available for dynamic scaling policies and when to use them (more detail further down the page):

Scaling Policy	**What it is**	**When to use**
Target Tracking Policy	Adds or removes capacity as required to keep the metric at or close to the specific target value.	You want to keep the CPU usage of your ASG at 70%
Simple Scaling Policy	Waits for the health check and cool down periods to expire before re-evaluating	Useful when load is erratic. AWS recommends step scaling instead of simple in most cases.
Step Scaling Policy	Increases or decreases the configured capacity of the Auto Scaling group based on a set of scaling adjustments, known as step adjustments.	You want to vary adjustments based on the size of the alarm breach

The diagram below depicts an Auto Scaling group with a Scaling policy set to a minimum size of 1 instance, a desired capacity of 2 instances, and a maximum size of 4 instances: