**What Is AWS Auto Scaling?**

AWS Auto Scaling enables you to configure automatic scaling for the AWS resources that are part of your application in a matter of minutes. You can configure automatic scaling for individual resources or for whole applications.

With AWS Auto Scaling, you configure and manage scaling for your resources through a scaling plan. The scaling plan uses dynamic scaling and predictive scaling to automatically scale your application’s resources. This ensures that you add the required computing power to handle the load on your application and then remove it when it's no longer required. The scaling plan lets you choose scaling strategies to define how to optimize your resource utilization. You can optimize for availability, for cost, or a balance of both. Alternatively, you can create custom scaling strategies.

AWS Auto Scaling is useful for applications that experience daily or weekly variations in traffic flow, including the following:

* Cyclical traffic such as high use of resources during regular business hours and low use of resources overnight
* On and off workload patterns, such as batch processing, testing, or periodic analysis
* Variable traffic patterns, such as marketing campaigns with periods of spiky growth

**Features of AWS Auto Scaling**

Use AWS Auto Scaling to automatically scale the following resources:

* **Amazon EC2 Auto Scaling groups**: Launch or terminate EC2 instances in an Auto Scaling group.
* **Amazon EC2 Spot Fleet requests**: Launch or terminate instances from a Spot Fleet request, or automatically replace instances that get interrupted for price or capacity reasons.
* **Amazon ECS**: Adjust the ECS service desired count up or down in response to load variations.
* **Amazon DynamoDB**: Enable a DynamoDB table or a global secondary index to increase or decrease its provisioned read and write capacity to handle increases in traffic without throttling.
* **Amazon Aurora**: Dynamically adjust the number of Aurora read replicas provisioned for an Aurora DB cluster to handle changes in active connections or workload.

The scaling features currently available are dynamic scaling and predictive scaling.

Dynamic scaling creates target tracking scaling policies for the scalable resources in your application. This lets your scaling plan add and remove capacity for each resource as required to maintain resource utilization at the specified target value. The default scaling metrics provided are based on the most commonly used metrics used for automatic scaling.

How predictive scaling works:

* **Load forecasting**: AWS Auto Scaling analyzes up to 14 days of history for a specified load metric and forecasts the future demand for the next two days. This data is available in one-hour intervals and updated daily.
* **Scheduled scaling actions**: AWS Auto Scaling schedules the scaling actions that proactively add and remove resource capacity to reflect the load forecast. At the scheduled time, AWS Auto Scaling updates the resource's minimum capacity with the value specified by the scheduled scaling action. The intention is to maintain resource utilization at the target value specified by the scaling strategy. If your application requires more capacity than is forecast, dynamic scaling is available to add additional capacity.
* **Maximum capacity behavior**: Each resource has a minimum and a maximum capacity limit between which the value specified by the scheduled scaling action is expected to lie. However, you can control whether your application can add resources beyond their maximum capacity when the forecast capacity is higher than the maximum capacity.

Currently, predictive scaling is only available for Amazon EC2 Auto Scaling groups.

## Related Services

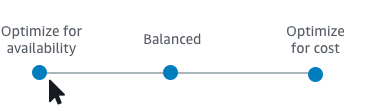
[AWS CloudFormation](https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/) allows you to use templates, which are formatted text files in JSON or YAML, to model and provision a collection of related AWS resources. You can also create templates of scaling plans using AWS CloudFormation.

[Amazon CloudWatch](https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/) is a monitoring service for AWS Cloud resources and the applications you run on AWS. CloudWatch lets you collect and track metrics, log files, and automatically react to changes in your applications using alarms.

**How AWS Auto Scaling Works**

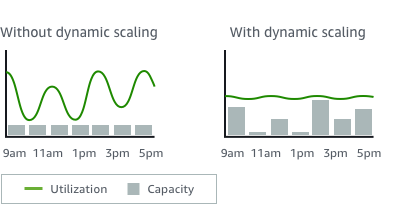
**What is a scaling strategy?**

The scaling strategy tells AWS Auto Scaling how to optimize the utilization of the resources in your scaling plan. You can optimize for availability, for cost, or a balance of both. Alternatively



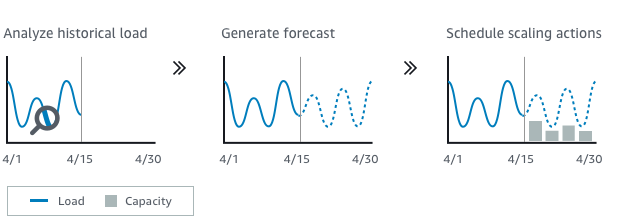
**What is dynamic scaling?**

Dynamic scaling creates target tracking scaling policies for the resources in your scaling plan. These scaling policies adjust resource capacity in response to live changes in resource utilization. The intention is to provide enough capacity to maintain utilization at the target value specified by the scaling strategy. This is similar to the way that your thermostat maintains the temperature of your home. You choose the temperature and the thermostat does the rest.



**What is predictive scaling?**

Predictive scaling uses machine learning to analyze each resource's historical workload and regularly forecasts the future load for the next two days, similar to how weather forecasts works. Using the forecast, it generates scheduled scaling actions to make sure that the resource capacity is available before your application needs it. Like dynamic scaling, predictive scaling works to maintain utilization at the target value specified by the scaling strategy.



For example, you can enable predictive scaling and configure your scaling strategy to keep the average CPU utilization of your Auto Scaling group at 50 percent. Your forecast calls for traffic spikes to occur every day at 8 o'clock in the morning. Your scaling plan creates the future scheduled scaling actions to make sure that your Auto Scaling group is ready to handle the traffic ahead of time. This helps keep the application performance constant, with the aim of always having the capacity required to maintain resource utilization as close to 50 percent as possible at all times.

# Advanced Settings (Optional)

* **Disable scale-in**—If this setting is cleared, automatic scale-in to decrease the current capacity of the resource is allowed when the specified metric is below the target value.
* **Cooldown**—Creates scale-out and scale-in cooldown periods. Cooldown periods are the amount of time after a scale-out or scale-in activity completes before another activity can start. The intention is to give newly provisioned resources time to start handling demand before triggering a new scaling action. This setting is not available if the resource is an Auto Scaling group. For more information, see [Cooldown Period](https://docs.aws.amazon.com/autoscaling/application/userguide/application-auto-scaling-target-tracking.html#target-tracking-cooldown) in the Application Auto Scaling User Guide.
* **Instance warmup**—[Auto Scaling groups only] Controls the amount of time that elapses before a newly launched instance begins contributing to the CloudWatch metrics. For more information, see [Instance Warmup](https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-scaling-target-tracking.html#as-target-tracking-scaling-warmup) in the Amazon EC2 Auto Scaling User Guide.
* **Predictive scaling mode**—Specifies the scaling mode. The default is **Forecast and scale**. If you change it to **Forecast only**, the scaling plan forecasts future capacity but doesn't apply the scaling actions.
* **Pre-launch instances**—Adjusts the scaling actions to run earlier when scaling out. For example, the forecast says to add capacity at 10:00 AM, and the buffer time is 5 minutes (300 seconds). The run time of the corresponding scaling action is then 9:55 AM. This is helpful for Auto Scaling groups, where it can take a few minutes from the time an instance launches until it comes in service. The actual time can vary as it depends on several factors, such as the size of the instance and whether there are startup scripts to complete. The default is 300 seconds.
* **Max capacity behavior**—Controls whether the selected resource can scale up above the maximum capacity when the forecast capacity is close to or exceeds the currently specified maximum capacity. The default is **Enforce the maximum capacity setting**.
  + **Enforce the maximum capacity setting**—AWS Auto Scaling cannot scale resource capacity higher than the maximum capacity. The maximum capacity is enforced as a hard limit.
  + **Set the maximum capacity to equal forecast capacity**—AWS Auto Scaling can scale resource capacity higher than the maximum capacity to equal but not exceed forecast capacity.
  + **Increase maximum capacity above forecast capacity**—AWS Auto Scaling can scale resource capacity higher than the maximum capacity by a specified buffer value. The intention is to give the target tracking scaling policy extra capacity if unexpected traffic occurs.
* **Max capacity behavior buffer**—If you chose **Increase maximum capacity above forecast capacity**, choose the size of the capacity buffer to use when the forecast capacity is close to or exceeds the maximum capacity. The value is specified as a percentage relative to the forecast capacity. For example, with a 10 percent buffer, if the forecast capacity is 50, and the maximum capacity is 40, then the effective maximum capacity is 55.

# What Is Application Auto Scaling?

Application Auto Scaling is a web service for developers and system administrators who need a solution for automatically scaling their scalable resources for individual AWS services beyond Amazon EC2. Application Auto Scaling allows you to configure automatic scaling for the following resources:

* Amazon ECS services
* Spot Fleet requests
* Amazon EMR clusters
* AppStream 2.0 fleets
* DynamoDB tables and global secondary indexes
* Aurora replicas
* Amazon SageMaker endpoint variants
* Custom resources provided by your own applications or services.

You have several options for scaling with AWS. For information about scaling your fleet of Amazon EC2 instances, see the [Amazon EC2 Auto Scaling User Guide](https://docs.aws.amazon.com/autoscaling/ec2/userguide/).

You can also use Application Auto Scaling and Amazon EC2 Auto Scaling in combination with AWS Auto Scaling to scale resources across multiple services. AWS Auto Scaling can help you maintain optimal availability and performance by combining predictive scaling and dynamic scaling (proactive and reactive approaches, respectively) together to scale your Amazon EC2 capacity faster. For more information, see the [AWS Auto Scaling User Guide](https://docs.aws.amazon.com/autoscaling/plans/userguide/).

## Features of Application Auto Scaling

Application Auto Scaling allows you to automatically scale your scalable resources according to conditions that you define.

* **Target tracking scaling**—Scale a resource based on a target value for a specific CloudWatch metric.
* **Step scaling**— Scale a resource based on a set of scaling adjustments that vary based on the size of the alarm breach.
* **Scheduled scaling**—Scale a resource based on the date and time.

# Target Tracking Scaling Policies for Application Auto Scaling

## Considerations

Keep the following considerations in mind:

You can have multiple target tracking scaling policies for a scalable target, provided that each of them uses a different metric. The intention of Application Auto Scaling is to always prioritize availability, so its behavior differs depending on whether the target tracking policies are ready for scale out or scale in. It will scale out the scalable target if any of the target tracking policies are ready for scale out, but will scale in only if all of the target tracking policies (with the scale-in portion enabled) are ready to scale in.

## Cooldown Period

The scale-out cooldown period is the amount of time, in seconds, after a scale-out activity completes before another scale-out activity can start. While this cooldown period is in effect, the capacity added by the initiating scale-out event is calculated as part of the desired capacity for the next scale-out event. The intention is to continuously (but not excessively) scale out.

The scale-in cooldown period is the amount of time, in seconds, after a scale-in activity completes before another scale-in activity can start. This cooldown period is used to block subsequent scale-in events until it has expired. The intention is to scale in conservatively to protect your application's availability. However, if another alarm triggers a scale-out policy during the cooldown period after a scale-in event, Application Auto Scaling scales out your scalable target immediately.

# Step Scaling Policies for Application Auto Scaling

**Limits**

* You cannot create step scaling policies for DynamoDB tables and global secondary indexes.

## Scaling Adjustment Types

When a step scaling policy is performed, it changes the current capacity of your scalable target using the scaling adjustment specified in the policy. A scaling adjustment can't change the capacity of the scalable target above the maximum capacity or below the minimum capacity.

Application Auto Scaling supports the following adjustment types for step scaling policies:

* **ChangeInCapacity**—Increase or decrease the current capacity of the scalable target by the specified value. A positive value increases the capacity and a negative value decreases the capacity.

Example: If the current capacity is 3 and the adjustment is 5, then Application Auto Scaling adds 5 to the capacity for a total of 8.

* **ExactCapacity**—Change the current capacity of the scalable target to the specified value. Specify a positive value with this adjustment type.

Example: If the current capacity is 3 and the adjustment is 5, then Application Auto Scaling changes the capacity to 5.

* **PercentChangeInCapacity**—Increase or decrease the current capacity of the scalable target by the specified percentage. A positive value increases the capacity and a negative value decreases the capacity. If the resulting value is not an integer, Application Auto Scaling rounds it as follows:
  + Values greater than 1 are rounded down. For example, 12.7 is rounded to 12.
  + Values between 0 and 1 are rounded to 1. For example, .67 is rounded to 1.
  + Values between 0 and -1 are rounded to -1. For example, -.58 is rounded to -1.
  + Values less than -1 are rounded up. For example, -6.67 is rounded to -6.

Example: If the current capacity is 10 and the adjustment is 10 percent, then Application Auto Scaling adds 1 to the capacity for a total of 11.

With **PercentChangeInCapacity**, you can also specify the minimum amount to scale (using the MinAdjustmentMagnitude parameter). For example, suppose that you create a policy that adds 25 percent and you specify a minimum amount of 2. If the scalable target has a capacity of 4 and the scaling policy is performed, 25 percent of 4 is 1. However, because you specified a minimum increment of 2, Application Auto Scaling adds 2.

# Scheduled Scaling for Application Auto Scaling

# Service-Linked Roles for Application Auto Scaling

Application Auto Scaling uses service-linked roles for the permissions that it requires to call other AWS services on your behalf. A service-linked role is a unique type of AWS Identity and Access Management (IAM) role that is linked directly to an AWS service.

Service-linked roles provide a secure way to delegate permissions to AWS services because only the linked service can assume a service-linked role. For more information, see [Using Service-Linked Roles](https://docs.aws.amazon.com/IAM/latest/UserGuide/using-service-linked-roles.html) in the IAM User Guide.

You can delete the roles only after first deleting their related resources. This protects your resources because you can't inadvertently remove permissions to access the resources.