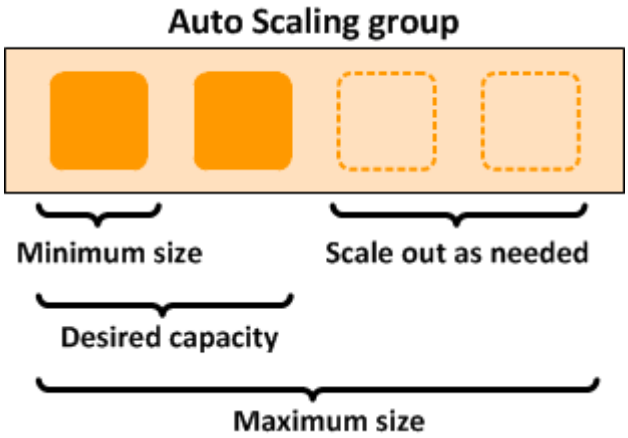
EC2 Autoscaling

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. You can specify the minimum number of instances in each Auto Scaling group, and Amazon EC2 Auto Scaling ensures that your group never goes below this size. You can specify the maximum number of instances in each Auto Scaling group, and Amazon EC2 Auto Scaling ensures that your group never goes above this size. If you specify the desired capacity, either when you create the group or at any time thereafter, Amazon EC2 Auto Scaling ensures that your group has this many instances. If you specify scaling policies, then Amazon EC2 Auto Scaling can launch or terminate instances as demand on your application increases or decreases.

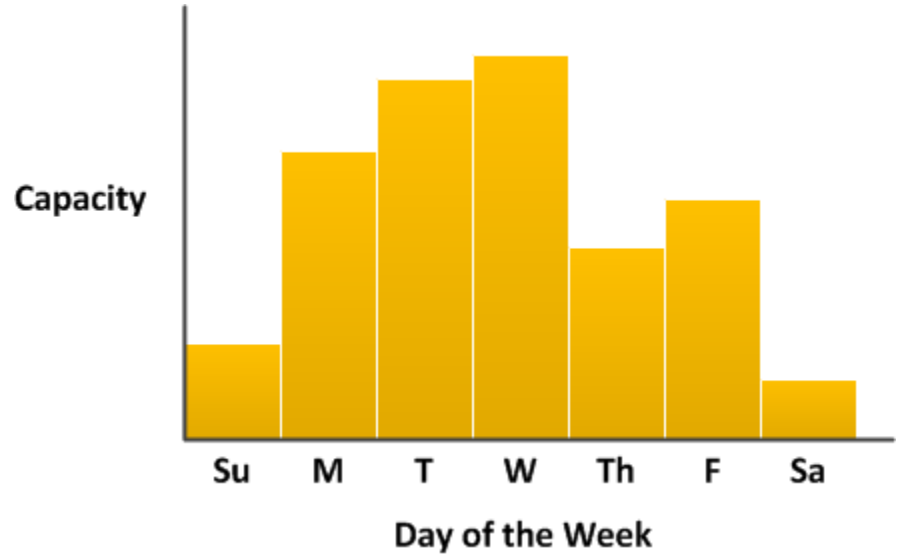
For example, the following Auto Scaling group has a minimum size of one instance, a desired capacity of two instances, and a maximum size of four instances. The scaling policies that you define adjust the number of instances, within your minimum and maximum number of instances, based on the criteria that you specify.



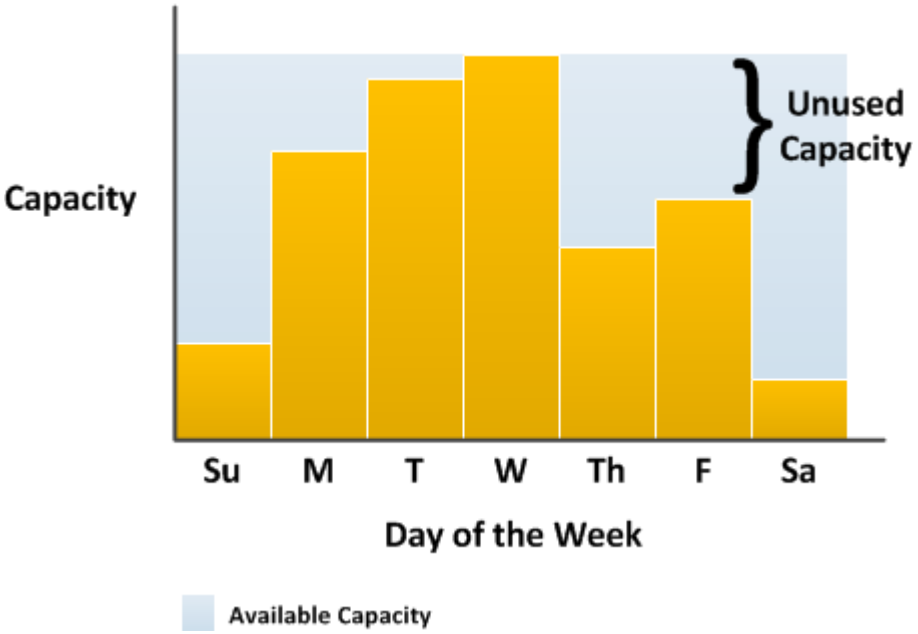
Usecase for Autoscaling

To demonstrate some of the benefits of Amazon EC2 Auto Scaling, consider a basic web application running on AWS. This application allows employees to search for conference rooms that they might want to use for meetings. During the beginning and end of the week, usage of this application is minimal. During the middle of the week, more employees are scheduling meetings, so the demand on the application increases significantly.

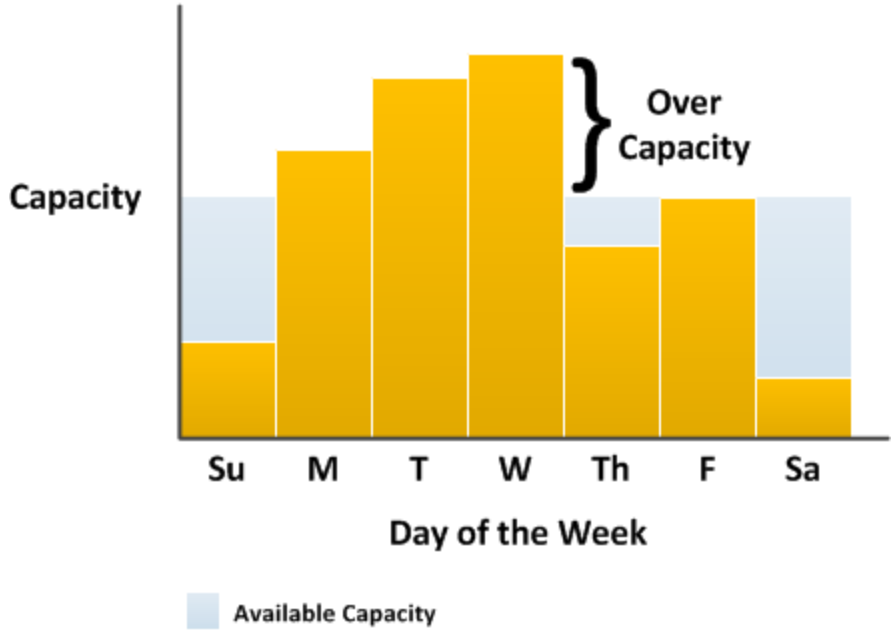
The following graph shows how much of the application's capacity is used over the course of a week.



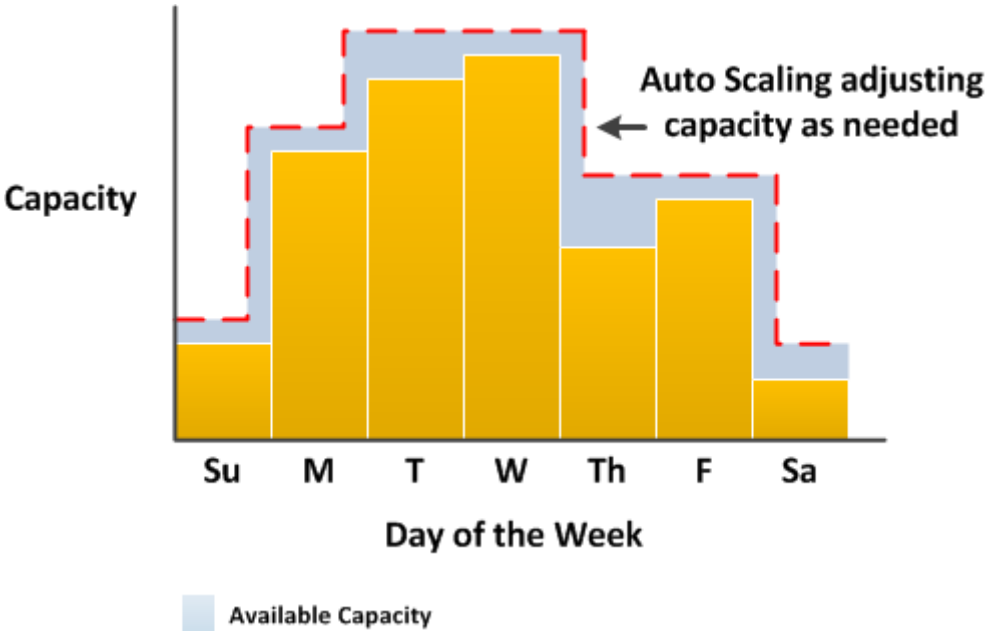
Traditionally, there are two ways to plan for these changes in capacity. The first option is to add enough servers so that the application always has enough capacity to meet demand. The downside of this option, however, is that there are days in which the application doesn't need this much capacity. The extra capacity remains unused and, in essence, raises the cost of keeping the application running.



The second option is to have enough capacity to handle the average demand on the application. This option is less expensive, because you aren't purchasing equipment that you'll only use occasionally. However, you risk creating a poor customer experience when the demand on the application exceeds its capacity.



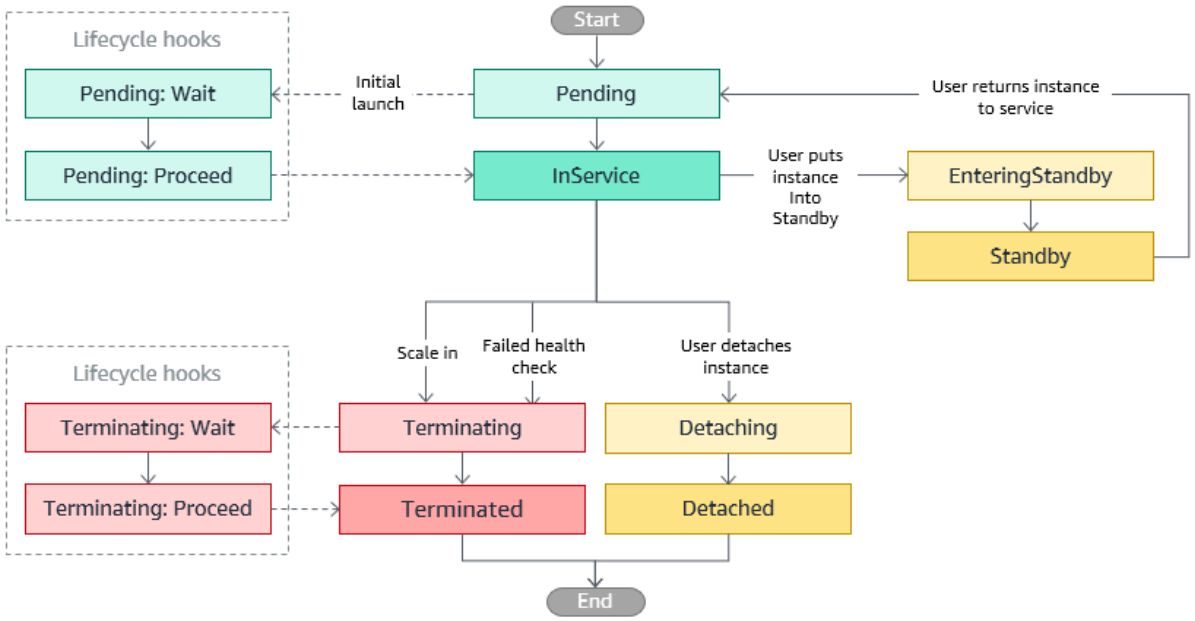
By adding Amazon EC2 Auto Scaling to this application, you have a third option available. You can add new instances to the application only when necessary, and terminate them when they're no longer needed. Because Amazon EC2 Auto Scaling uses EC2 instances, you only have to pay for the instances you use, when you use them. You now have a cost-effective architecture that provides the best customer experience while minimizing expenses.



Amazon EC2 Auto Scaling instance lifecycle

The EC2 instances in an Auto Scaling group have a path, or lifecycle, that differs from that of other EC2 instances. The lifecycle starts when the Auto Scaling group launches an instance and puts it into service. The lifecycle ends when you terminate the instance, or the Auto Scaling group takes the instance out of service and terminates it.

The following illustration shows the transitions between instance states in the Amazon EC2 Auto Scaling lifecycle.



Scale Out

The following scale-out events direct the Auto Scaling group to launch EC2 instances and attach them to the group:

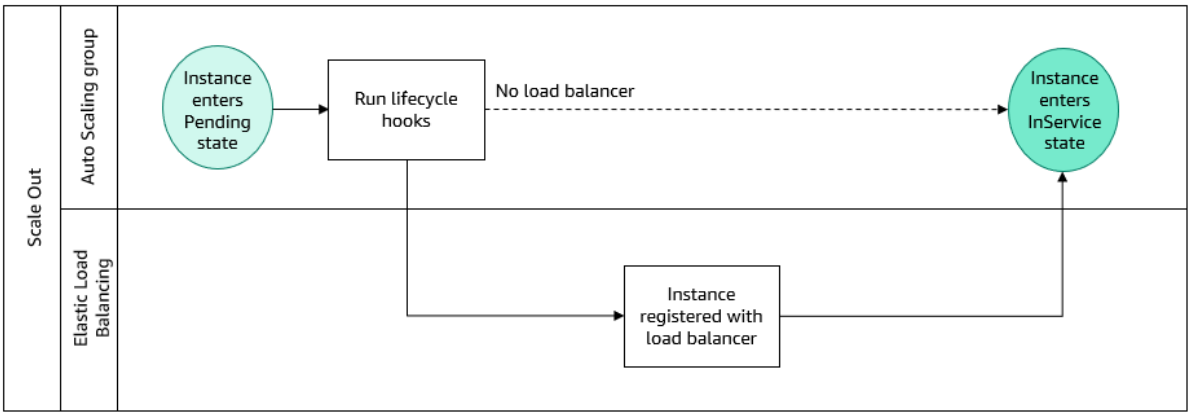
- You manually increase the size of the group.
- You create a scaling policy to automatically increase the size of the group based on a specified increase in demand.
- You set up scaling by schedule to increase the size of the group at a specific time.

When a scale-out event occurs, the Auto Scaling group launches the required number of EC2 instances, using its assigned launch template. These instances start in the `Pending` state. If you add a lifecycle hook to your Auto Scaling group, you can perform a custom action here.

When each instance is fully configured and passes the Amazon EC2 health checks, it is attached to the Auto Scaling group and it enters the `InService` state. The instance is counted against the desired capacity of the Auto Scaling group.

If your Auto Scaling group is configured to receive traffic from an Elastic Load Balancing load balancer, Amazon EC2 Auto Scaling automatically registers your instance with the load balancer before it marks the instance as `InService`.

The following summarizes the workflow for registering an instance with a load balancer for a scale-out event.



Instance in Service

Instances remain in the `InService` state until one of the following occurs:

- A scale-in event occurs, and Amazon EC2 Auto Scaling chooses to terminate this instance in order to reduce the size of the Auto Scaling group.
- You put the instance into a `Standby` state.
- You detach the instance from the Auto Scaling group.
- The instance fails a required number of health checks, so it is removed from the Auto Scaling group, terminated, and replaced.

Scale In

The following scale-in events direct the Auto Scaling group to detach EC2 instances from the group and terminate them:

- You manually decrease the size of the group
- You create a scaling policy to automatically decrease the size of the group based on a specified decrease in demand.
- You set up scaling by schedule to decrease the size of the group at a specific time.

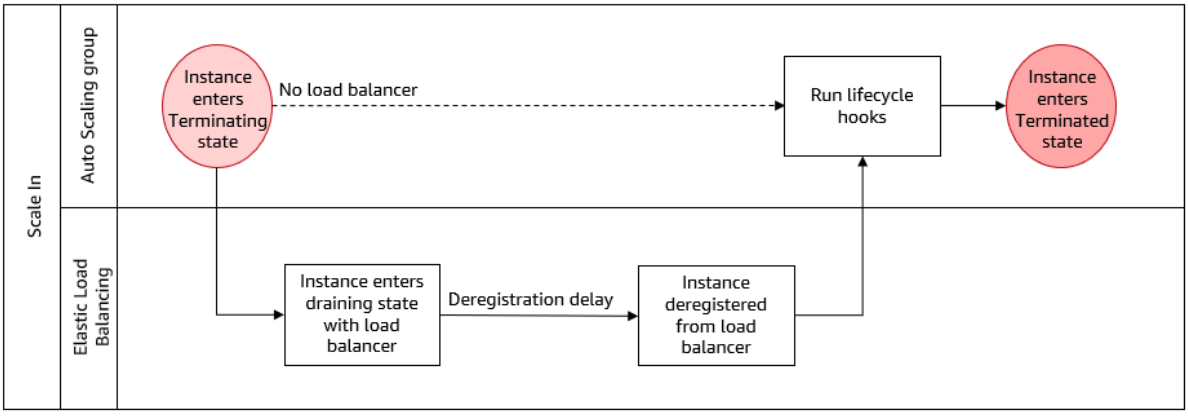
It is important that you create a corresponding scale-in event for each scale-out event that you create. This helps ensure that the resources assigned to your application match the demand for those resources as closely as possible.

When a scale-in event occurs, the Auto Scaling group terminates one or more instances. The Auto Scaling group uses its termination policy to determine which instances to terminate. Instances that are in the process of terminating from the Auto Scaling group enter the Terminating state, and can't be put back into service.

If your Auto Scaling group is configured to receive traffic from an Elastic Load Balancing load balancer, Amazon EC2 Auto Scaling automatically deregisters the terminating instance from the load balancer. Deregistering the instance ensures that all new requests are redirected to other instances in the load balancer's target group while existing connections to the instance are allowed to continue until the deregistration delay expires.

If you add a lifecycle hook to your Auto Scaling group, you can perform a custom action on the terminating instance. For more information, see Lifecycle hooks. Finally, the instance is completely terminated and enters the Terminated state.

The following summarizes the workflow for deregistering an instance with a load balancer for a scale-in event.



Amazon EC2 Auto Scaling lifecycle hooks

Amazon EC2 Auto Scaling offers the ability to add lifecycle hooks to your Auto Scaling groups. These hooks let you create solutions that are aware of events in the Auto Scaling instance lifecycle, and then perform a custom action on instances when the corresponding lifecycle event occurs. A lifecycle hook provides a specified amount of time (one hour by default) to wait for the action to complete before the instance transitions to the next state.

As an example of using lifecycle hooks with Auto Scaling instances:

When a scale-out event occurs, your newly launched instance completes its startup sequence and transitions to a wait state. While the instance is in a wait state, it runs a script to download and install the needed software packages for your application, making sure that your instance is fully ready before it starts receiving traffic. When the script is finished installing software, it sends the complete-lifecycle-action command to continue.

When a scale-in event occurs, a lifecycle hook pauses the instance before it is terminated and sends you a notification using Amazon EventBridge. While the instance is in the wait state, you can invoke an AWS Lambda function or connect to the instance to download logs or other data before the instance is fully terminated.

EC2 Autoscaling Group Scaling Policies

- EC2 Auto Scaling Policies provide several ways for scaling the Auto Scaling group.
  - Manual Scaling
  - Scheduled Scaling
  - Dynamic Scaling
  - Predictive Scaling

Manual Scaling

- Manual scaling can be performed by
  - Changing the desired capacity limit of the ASG
  - Attaching/Detaching instances to the ASG

Scheduled Scaling

- Scaling based on a schedule allows you to scale the application in response to **predictable load changes** *for e.g. last day of the month, the last day of a financial year.*
- Scheduled scaling requires the configuration of Scheduled actions, which tells Auto Scaling to perform a scaling action at a certain time in the future, with the

start time at which the scaling action should take effect, and the new minimum, maximum, and desired size of group should have.
- Auto Scaling guarantees the order of execution for scheduled actions within the same group, but not for scheduled actions across groups.

Dynamic Scaling

- Allows automatic scaling in response to the changing demand *for e.g. scale-out in case CPU utilization of the instance goes above 70% and scale in when the CPU utilization goes below 30%*
- ASG uses a combination of **alarms & policies** to determine when the conditions for scaling are met.
  - An alarm is an object that watches over a single metric over a specified time period. When the value of the metric breaches the defined threshold, for the number of specified time periods the alarm performs one or more actions (such as sending messages to Auto Scaling).
  - A policy is a set of instructions that tells Auto Scaling how to respond to alarm messages.
- Dynamic scaling process works as below
  1. CloudWatch monitors the specified metrics for all the instances in the Auto Scaling Group.
  2. Changes are reflected in the metrics as the demand grows or shrinks
  3. When the change in the metrics breaches the threshold of the CloudWatch alarm, the CloudWatch alarm performs an action. Depending on the breach, the action is a message sent to either the scale-in policy or the scale-out policy
  4. After the Auto Scaling policy receives the message, Auto Scaling performs the scaling activity for the ASG.
  5. This process continues until you delete either the scaling policies or the ASG.
- When a scaling policy is executed, if the capacity calculation produces a number outside of the minimum and maximum size range of the group, EC2 Auto Scaling ensures that the new capacity never goes outside of the minimum and maximum size limits.
- When the desired capacity reaches the maximum size limit, scaling out stops. If demand drops and capacity decreases, Auto Scaling can scale out again.

Predictive Scaling

- Predictive scaling can be used to increase the number of EC2 instances in the ASG in advance of daily and weekly patterns in traffic flows.
- Predictive scaling is well suited for situations where you have:
  - Cyclical traffic, such as high use of resources during regular business hours and low use of resources during evenings and weekends
  - Recurring on-and-off workload patterns, such as batch processing, testing, or periodic data analysis

    - Applications that take a long time to initialize, causing a noticeable latency impact on application performance during scale-out events
- Predictive scaling provides proactive scaling that can help scale faster by launching capacity in advance of forecasted load, compared to using only dynamic scaling, which is reactive in nature.
- Predictive scaling uses machine learning to predict capacity requirements based on historical data from CloudWatch. The machine learning algorithm consumes the available historical data and calculates the capacity that best fits the historical load pattern, and then continuously learns based on new data to make future forecasts more accurate.
- Predictive scaling supports **forecast only** mode so that you can evaluate the forecast before you allow predictive scaling to actively scale capacity
- When you are ready to start scaling with predictive scaling, switch the policy from **forecast only** mode to **forecast and scale** mode.