Load Balancing and Auto Scaling

Pooja Diwakar

Cloud Computing

UCSC Extension

[Amazon EC2’s Auto Scaling 2](#_Toc438065133)

[Example: Covering Variable Demand 3](#_Toc438065134)

[Azure’s Load Balancer and Auto Scaling 4](#_Toc438065135)

[Azure Pricing: 5](#_Toc438065136)

# Amazon EC2’s Auto Scaling

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

Auto Scaling helps you ensure that you have the correct number of 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 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 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, Auto Scaling ensures that your group has this many instances. If you specify scaling policies, then Auto Scaling can launch or terminate instances as demand on your application increases or decreases.

When you use Auto Scaling, you can automatically increase the number of EC2 instances you're using when the user demand goes up, and you can decrease the number of EC2 instances when demand goes down. As Auto Scaling dynamically adds and removes EC2 instances, you need to ensure that the traffic coming to your web application is distributed across all of your running EC2 instances. AWS provides the Elastic Load Balancing service to distribute the incoming web traffic (called the load) automatically among all the EC2 instances that you are running. Elastic Load Balancing manages incoming requests by optimally routing traffic so that no one instance is overwhelmed. Using Elastic Load Balancing with your auto-scaled web application makes it easy to route traffic across a dynamically changing fleet of EC2 instances.

Elastic Load Balancing uses load balancers to monitor traffic and handle requests that come through the Internet. Your load balancer acts as a single point of contact for all incoming traffic to the instances in your Auto Scaling group.

By default, the Auto Scaling group determines the health state of each instance by periodically checking the results of EC2 instance status checks. Elastic Load Balancing also performs health checks on the EC2 instances that are registered with the load balancer.

Adding Auto Scaling to your application architecture is one way to maximize the benefits of the AWS cloud. When you use Auto Scaling, your applications gain the following benefits:

* Better fault tolerance. Auto Scaling can detect when an instance is unhealthy, terminate it, and launch an instance to replace it.
* Better availability. You can configure Auto Scaling to use multiple Availability Zones. If one Availability Zone becomes unavailable, Auto Scaling can launch instances in another one to compensate.
* Better cost management. Auto Scaling can dynamically increase and decrease capacity as needed. Because you pay for the EC2 instances you use, you save money by launching instances when they are actually needed and terminating them when they aren't needed.

## Example: Covering Variable Demand

To demonstrate some of the benefits of 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 demands on the application increases significantly.

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 demands 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 demands on the application exceeds its capacity.

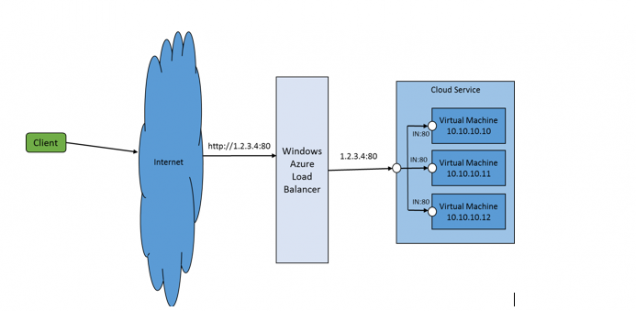
By adding Auto Scaling to this application, you have a third option available. You can add new instances to the application only when necessary, and terminate them when they're no longer needed. And because Auto Scaling uses EC2 instances, you only have to pay for the instances you use, when you use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

# Azure’s Load Balancer and Auto Scaling

Microsoft Azure offers load balancing services for virtual machines (IaaS) and cloud services (PaaS) hosted in the Microsoft Azure cloud. Load balancing allows your application to scale and provides resiliency to application failures among other benefits.

The load balancing services can be accessed by specifying input endpoints on your services either via the Microsoft Azure Portal or via the [service model](http://msdn.microsoft.com/en-us/library/windowsazure/ee758711.aspx) of your application. Once a hosted service with one or more input endpoints is deployed in Microsoft Azure, it automatically configures the load balancing services offered by Microsoft Azure platform. To get the benefit of resiliency / redundancy of your services, you need to have at least two virtual machines serving the same endpoint.

The following diagram is an example of an application hosted in Microsoft Azure that uses load balancing service to direct incoming traffic (on address/port 1.2.3.4:80) to three virtual machines, all listening on port 80.

[](http://acom.azurecomcdn.net/80C57D/blogmedia/blogmedia/2014/04/azureloadbalance.png)

You can configure your cloud service to automatically increase or decrease the number of instances or Virtual Machines that are used by your application. You can configure scaling based on the following parameters:

* [Average CPU usage](https://azure.microsoft.com/en-us/documentation/articles/cloud-services-how-to-scale/#averagecpu) - If the average percentage of CPU usage goes above or below specified thresholds, role instances are created or deleted, or Virtual Machines are turned on or turned off from an availability set.
* [Queue messages](https://azure.microsoft.com/en-us/documentation/articles/cloud-services-how-to-scale/#queuemessages) - If the number of messages in a queue goes above or below a specified threshold, role instances are created or deleted, or Virtual Machines are turned on or turned off from an availability set.

## Azure Pricing:

For an 8 core Virtual Machine, Azure charges $0.616/ hour.

So if we consider an online business that allows customers to shop online, the shopping is going to peak during holiday season. If to handle the peaks, 1000 virtual machines are required, then the business needs to look at the following:

Scenario 1: 12 months

Holiday Season – 2 months

1000 Virtual Machines at $0.616 per hour = $616 per hour

**Total = $616 \* 24 \* 60 = $887,040**

So the business has to pay the above total to keep the 1000 virtual machines to handle the peaks.

Scenario 2:

But if 400 virtual machines are required to handle regular traffic and the peak traffic is aggregated to be equivalent to 7 days (including Thanksgiving and Christmas) then we get the following:

400 Virtual Machines at $0.616 per hour = $246.4 per hour

**Total = $246.4 \* 24 \* 53 = $313,420**

For peaks if Auto Scaling, increases the number of instances to 1000 for an aggregate of 7 days:

1000 Virtual Machines at $0.616 per hour = $616 per hour

**Total = $616 \* 24 \* 7 = $103,488**

**Grand total = $313,420 + $103,488 = $416908**

We see that the auto scaling functionality is cutting the cost almost half. Online businesses can thus leverage this functionality to create cost effective, highly available, cloud based solutions.

Sources:

1. <http://docs.aws.amazon.com/AutoScaling/latest/DeveloperGuide/how-as-works.html>
2. <https://azure.microsoft.com/en-us/blog/microsoft-azure-load-balancing-services/>
3. <https://azure.microsoft.com/en-us/pricing/details/virtual-machines/>