Azure – Scaling Applications

One of the major benefits of using the cloud is scalability. With Azure auto scaling, you can scale up and out like you couldn’t do with your own hardware, as much as your pay capacity. And importantly, you can scale down and in when you don’t need the resources, thereby saving money. This would not be possible if you bought servers on-premises to accommodate your peak load.

There are two main ways to scale resources:

* Vertical: Scaling up and down
* Horizontal: Scaling out and in

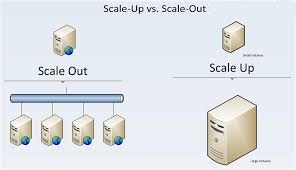
## **Scaling an application**

Scaling-up refers to increasing the compute power of the hosting nodes i.e. increase the capacity of the servers by increasing memory, processing power, or drive spaces. Scaling-down is opposite, decrease capacity of server. Scaling-up has certain constraints as physical machines only support certain memory and disk.

Scaling-out is a horizontal approach. Instead of trying to increase the compute power of existing nodes, scaling-out brings in more hosting nodes to share the workload. There’s no theoretical limit to how much you can scale-out, you can add as many nodes as needed. This makes it possible for an application to be scaled to very high capacity that is often hard to achieve with scaling-up. Scaling-in is opposite, decrease number of instances that application runs on.

Scaling-out is a preferable scaling method for cloud applications.

Application can be scaled manually or automatically, Auto-scaling is a way to automatically scale up/down, in/out the number of compute resources that are being allocated to application based on its needs at any given time.

Scaleup vs Scaleout

|  |  |
| --- | --- |
| Scale Out | Scale up |
| Add more of the same servers | Add more resource to an existing server e.g. cores, RAM, Disk space |
| More difficult to scale existing application | Easier to scale existing application |
| More cost effective for large scale applications | Limited by cost and physics |
| Likely to need infra and application change |  |

## **Why auto scale applications?**

* Better fault tolerance. Auto Scaling can detect when an instance is unhealthy, terminate it, and launch an instance to replace it.
* Better availability. Auto Scaling can help you ensure that your application always has the right amount of capacity to handle the current traffic demands.
* Better cost management. Auto Scaling can dynamically increase and decrease capacity as needed. Because you pay for the instances you use, you save money by launching instances when they are actually needed and terminating them when they aren’t needed.

## **Key areas to consider for scaling applications**

Scaling an application is a complex problem that does not have a “one size fits all” solution. Simply adding resources to a system or running more instances of a process doesn’t guarantee that the performance of the system will improve.  To correctly scale your application there are few key areas that will contribute to applications success:

**1.** Understanding application architecture and its weaknesses.  
Is Application Stateful? Stateless?  
What are all the components of application?  
Where are the bottlenecks in the application?  
When load is applied to app, what will break first?

**2.**  Understanding the expected load and performance requirements.  
Does the application need to serve one thousand users? Or one million?  
Will traffic come from a single geographic location or globally?  
Are there seasonal variations? Traffic peaks?  
How fast should the app respond? 1 second? 1 millisecond?

**3.** Understanding and correctly leverage the platform hosting.  
What features should be leveraged to achieve scale goals?

**4.** Consider [Pipes and Filters Pattern](https://docs.microsoft.com/en-us/azure/architecture/patterns/pipes-and-filters)If the solution implements a long-running task, design this task to support both scaling out and scaling in.

**5.** Consider throttling the services  
Auto scale takes some time to provision hardware, but in case of sudden burst of workload services might break by the time. See the [Throttling Pattern](https://docs.microsoft.com/en-us/azure/architecture/patterns/throttling).

## **Auto scale Azure solutions**

Azure provides built-in auto scaling for following compute options.

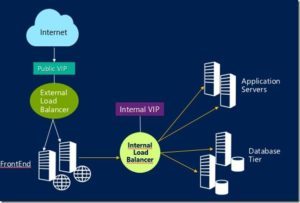
* Virtual Machines support auto scaling through the use of [VM Scale Sets](https://docs.microsoft.com/en-us/azure/virtual-machine-scale-sets/virtual-machine-scale-sets-overview), which are a way to manage a set of Azure virtual machines as a group.
* Service Fabric supports auto-scaling through VM Scale Sets. Every node type in a Service Fabric cluster is set up as a separate VM scale set.
* Azure App Service has built-in auto scaling. Auto scale settings apply to all of the apps within an App Service.
* Azure Cloud Services has built-in auto scaling at the role level.
* Azure Functions automatically allocates compute power when code is running, scaling out as necessary to handle load.

## **Workload Distribution**

When an application is scaled-out, the workload needs to be distributed among the participating instances. This is done by load balancing, Traffic Manager, Application Gateway in Azure.

Load Balancer  
Applications are generally designed in multi-tier architecture. Hence the application workload needs to be distributed among the participating instances by the Azure public-facing load-balancer and middle tiers and database tiers that aren’t directly accessible from the Internet. Azure has introduced Internal Load Balancers (ILB) to provide load balancing among VMs residing in a Cloud Service or a regional virtual network.

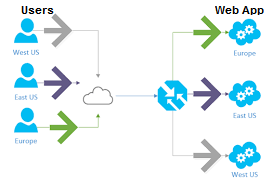
End users access the presentation layer. The requests are distributed to the presentation layer VMs by Azure Load Balancer. Then, the presentation layer accesses the database servers through an internal load balancer.

Load Balancer

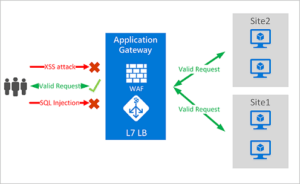
Azure Traffic Manager  
The job of Azure Traffic Manager is to route traffic globally based on flexible policies, enabling an excellent user experience that aligns with how you’ve structured your application across the world. Traffic Manager works at the DNS level. It uses DNS responses to direct end-user traffic to globally distributed endpoints. Clients then connect to those endpoints directly.

Traffic Manager has several different policies:

* Performance routing to send the requestor to the closest endpoint in terms of latency.
* Priority routing to direct all traffic to an endpoint, with other endpoints as backup.
* Weighted round-robin routing, which distributes traffic based on the weighting that is assigned to each endpoint.

Traffic Manager

Application Gateway  
Microsoft Azure Application Gateway offers various layer 7 load balancing capabilities for application. It allows customers to optimize web farm productivity by offloading CPU intensive SSL termination to the application gateway. It also provides other layer 7 routing capabilities including round robin distribution of incoming traffic, cookie-based session affinity, URL path-based routing, and the ability to host multiple websites behind a single Application Gateway. A web application firewall is also provided as part of the application gateway.

Application Gateway

## **Scale Azure Web Application**

Azure provides scale up and scale out scaling option for azure web apps and some other resources. You can either opt for manual scale or auto scale options.

In manual scale option you need to either login to portal, manually change the settings of the web app to upgrade or downgrade the instances and pricing tier of the web app, or setup the auto scale rules based on the CPU usage, Memory usage, concurrent http requests etc.

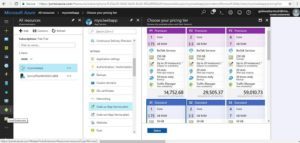
Scale up Web application

Let’s consider a scenario where you have created a web app and you had selected standard pricing tier at the time of creation. Over a period of time, traffic on your web app increased and now you want to add the infrastructure to the web app so that it provides better performance.

You can do so by changing the pricing tier of the web app.

Login to the portal and select the web app

In the settings blade select scale up option and select premium pricing tier which provided better hardware options than standard tier.

Scale up Azure Web App

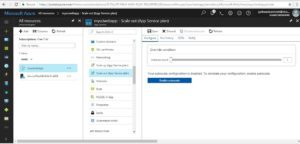
### **Scale out Web Application**

Another option azure provides for scaling web app is scale out, in case you have a business scenario in which traffic on your web site increase exponentially at certain time and performance of the site goes down during that time, to address this issue you can opt for scale out option.

There are two ways you can scale out web app

Manual

You can select multiple instances of the web app, this will be hard coded number of instances and will remain same irrespective of traffic on the site, and with this option you would be losing the computation power during low traffic on site, and might be in crunch of computation power during peak traffic on the web app. So solve this issue we have auto scale option.

Manual – Scale out Azure Web App

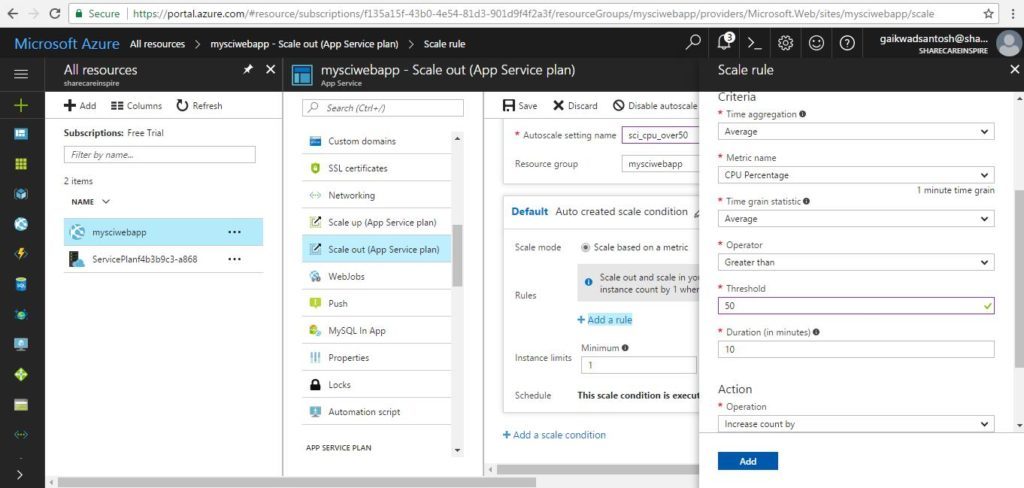
Auto scale

Auto scale option provides flexibility to add computation power to web whenever required, and reduce the computation when not required.

You can set the scale rules to increase and decrease instances based on the performance indicators like CPU usage, Memory usage, HTTP requests etc.

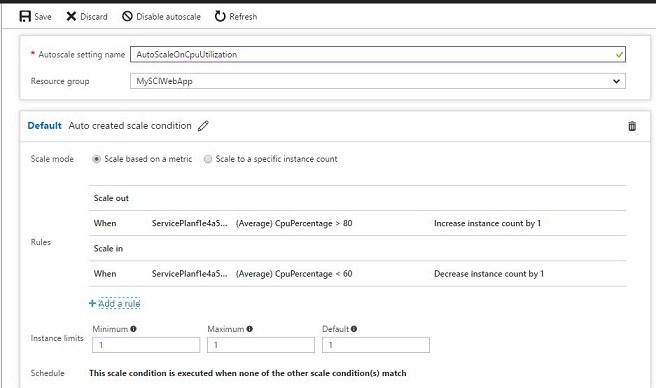
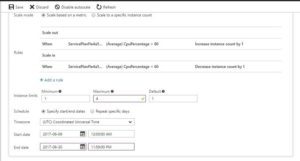
This option helps to provide better services (performance) to the end users and also save the azure billing cost.

There are certain design principles need to be considered while developing web apps which are planned for scale out (e.g. state management).

Auto – Scale out Azure Web App

In above screen snap Web app is configured to scale out by 1 instance when CPU utilization reaches to 80% threshold.

You can further add more rules to scale condition for scale in and scale out, like if CPU utilization goes beyond 80 then increase instances by 1 and if CPU utilization falls below 60% then reduce instance by 1 as follows.

Scale out app by 1 instance if CPU utilization is grater than 80 and decrease by 1 when less than 60. you can also schedule scale out condition to run for specific period.Schedule Scale out condition

Following are certain problems you might face during scaling out web app, and you would need to make changes to the web app.

* File access  
  If web app is reading/writing files, then file written to disks cannot be accessible to other nodes at same time, hence need to take care of concurrent requests to same resource.
* Session State  
  If web app is using in memory session state, it will not be available to other nodes. you will need to configure an external session state provider (either the Redis Cache Service or a SQL Server session state provider).
* Caching  
  In memory cache needs distribution to other nodes
* Bottlenecks  
  if single Database gets requests by multiple business layer components at a time, need to be handled. consider using Elastic DB service provided by Azure which provide option of data Sharding & Scale out.

Auto scaling Azure Websites is an important strategy for providing a good experience to website users, also helps in reducing overall costs. it also simplifies the setup, configuration and maintenance of the website, it can allow for developers to move faster focusing on the essentials of the website experience.