REF

<https://www.concurrency.com/blog/w/azure-traffic-manager-vs-azure-load-balancer>

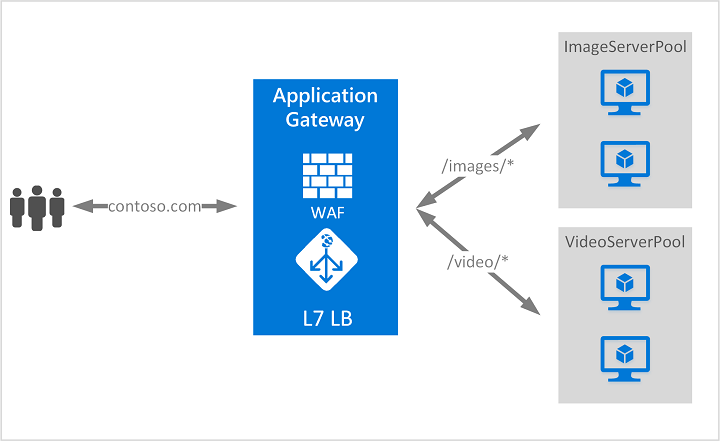
<http://rahulrajatsingh.com/2018/07/understanding-azure-load-balancing-solutions-azure-load-balancer-azure-application-gateway-and-azure-traffic-manager/>

<https://adityaswami89.wordpress.com/2016/12/21/difference-between-azure-balancer-application-gateway-and-traffic-manager/>

**Application Gateway**

The IP of the application gateway can be public or private

For example, you can route traffic based on the incoming URL. So if /images is in the incoming URL, you can route traffic to a specific set of servers (known as a pool) configured for images. If /video is in the URL, that traffic is routed to another pool optimized for videos



Features:

1. Auto scaling
2. Custom error pages - Application Gateway allows you to create custom error pages instead of displaying default error pages
3. URL based routing - For example, requests for http://contoso.com/video/\* are routed to VideoServerPool, and http://contoso.com/images/\* are routed to ImageServerPool. DefaultServerPool is selected if none of the path patterns match.
4. Multiple site hosting - enables you to configure more than one web site on the same application gateway instance. This feature allows you to configure a more efficient topology for your deployments by adding up to 100 web sites to one application gateway. Each web site can be directed to its own pool. For example, application gateway can serve traffic for contoso.com and fabrikam.com from two server pools called ContosoServerPool and FabrikamServerPool.
5. Requests for http://contoso.com are routed to ContosoServe rPool, and http://fabrikam.comare routed to FabrikamServerPool

* Azure Application Gateway only supports URL and Path based routing controls
* Gateway monitors the health of each back-end resource and tracks its health rating. It automatically removes a resource from the pool if the resource health rating dips below a predefined level
* Application Gateway is available in all regions of global Azure.
* Only one public IP address is supported on an application gateway

**Endpoint**

In case we need them to communicate with our own computer, we will need an endpoint configured to make it happen. It is basically accessing the virtual machine through a port. An endpoint provides remote access to the services running on virtual machine. It has a public and private port that needs to be specified while creating an endpoint

**Traffic Manager**

If you have resources at different locations and depending the end user’s  closest latency you wish to direct the traffic to an end point, you should use Traffic Manager

* Latency.  Direct to the "closest service"
* Round Robin.  Distribute across all services
* Failover.  Direct to backup if primary fails
* Nested.  Flexible multi-level policies

The client makes a DNS request and, based on the location of the DNS, Azure Traffic Manager will find the nearest region and sends that back to the client via a DNS response

**Load Balancer**

The job of **Azure Load Balancer is to direct traffic inside a region**

The load balancer is configured with load balancing rules and it these rules work at the port level. It accepts source and destination ports map them together so that whenever it receives a request for the source port, the request is forwarded to a virtual machine from a group of virtual machines (or application in VNET) attached to the load balancer on the destination port.

Azure load balancer can be used in two configuration modes:

* **External**– Public load balancing
* **Internal**– Internal load balancing

#### External – Public load balancing

In this mode, load balancer is assigned a public IP address to ensure that the load balancer can accept requests coming in from the internet. The load balancer will get called from the internet by the client applications and services, and then based on the configured rules it will distribute the incoming traffic over VMs, containers, or apps.

#### Internal – Internal load balancing

The internal Load Balancer is essentially the same as external, but it uses a private IP address and thus it can be called only from the applications within the the virtual network to which it is attached.

Example:

* Create two VM’s with availability set
* Create load balancer
* Load balancer always has another IP address
* Any request comes to load balancer, it will send request to both VM based on load balancer algorithm
* In load balancer, backend pool add the both VM’s
* Add health probes in load balancer
* Then if you hit the load balancer IP address, it will redirect to one of the VM

**Availability Set**

Availability set is a great feature Azure provides for Business continuity. So when two or more servers are in a single availability set, even if one or more of servers becomes unavailable due to any reason, rest of servers in an Availability set will provide the service.

For example: Consider I have hosted my site in two web servers say Web1 and Web2 which are in availability set. If Web1 becomes unavailable due to any reason for some time, Web2 will provide the service.

Also, Microsoft offers 99.99% uptime for the servers in Availability set.

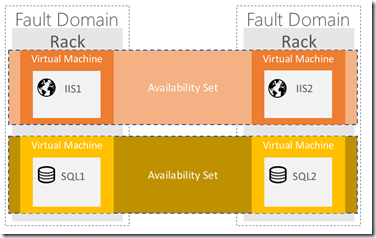
You use an availability set when you have a predictable workload that you want to protect against downtime. In other words, if you know your solution can run on a single VM, you would create two VMs in an availability set to ensure that at least one VM is running at all times, and a load balancer to route traffic to an active VM.

You use **scale sets** when the workload on your solution changes a lot or is unpredictable. For example, if you have a solution that normally works using only one VM, but may need up to 5 VMs to work under heavy demand, you would use a scale set to ensure that your solution quickly scales to demand. Scale sets pre-provision your VM instances so that when they need to scale out, it's just a matter of starting them, rather than having to wait for Azure to actually create the VMs.

Fault and update domains are automatically assigned when you place two or more virtual machines in an availability set. So, when you create your VMs, just provide the name of the availability set to which they should be assigned, and Azure will take care of the rest.

There are 3 fault domains and 5 update domains per availability set, by default.

When you put VMs in to an availability set, Azure guarantees to spread them across Fault Domains and Update Domains. A Fault Domain (FD) is essentially a rack of servers. It consumes subsystems like network, power, cooling etc. So 2 VMs in the same availability set means Azure will provision them in to 2 different racks so that if say, the network or the power failed, only one rack would be affected.



Sometimes you need to update your app, or Microsoft needs to update the host on which your VM(s) are running. Note that with IaaS VMs, Microsoft does not automatically update your VMs. You have complete control (and responsibility) over that. But say if a serious security vulnerability is identified and a patch created. It’s in Microosft’s interest to get that applied to the host underneath your VM as soon as possible. So how is that done without taking your service offline? Update Domains. It’s similar to the FD methods, only this time, instead of an accidental failure, there is a purposeful move to take down one (or more) of your servers. So to make sure your service doesn’t go offline because of an update, it will walk through your update domains one after the other