Computer Services

We can view compute services as one of the technical pillars of Azure.

Technical pillars:

- Compute services
- Networking services
- Storage services
- Databased services

Note: These are infrastructure/foundational level services.

Compute Services

Basically, this is refereeing to executing code in the cloud. It could be a website, app, banking functionality for a bank (logging/check bank balance).

MS Azure dozens of Compute Types. Here are a few:

- Virtual Machines (VM)
- VM Scale Sets (VMSS)
- App services (Web apps)
- Azure Container Instances (ACI)
- Azure Container Apps
- Azure Kubernetes Service (AKS)
- Azure Virtual Desktop

Virtual Machines:

This is the closest analogue to a “server” (computer) in cloud computing. But it is virtual. That is, a single physical machine has been subdivide into slices and you get to rent a single slice of it.

Standalone Server Analogy:

Think of this as a single-detached house. You can do whatever you want with it (generally). It is very difficult to do anything that affects your neighbors (soundproof walls). You do not share any services with your neighbors (besides garbage, sewer, water, electricity – provided by the city).

Virtualization Analogy:

Think of a "host" as an apartment building on the same land. A "virtual machine" is an apartment in that building. You are using common services (garbage, sewer, water, electricity) and other services available (shared gym, heating/cooling). It is generally cheaper to rent (but you would have same the feeling as you in a house / get the benefits of a house).

Thus, virtual machines:

- Infrastructure as a service – IaaS
- Take an existing machine (desktop / server) from your environment into the cloud (a copy)
- Windows / Linus operating systems – several of each
- A "slice" of a physical machine shared with other customers
- Full control over it, as if it was your machine

Note: In AWS, a Virtual Machine is called Elastic Compute Cloud (EC2).

Virtual Machine Types:

You cannot arbitrary enter how many CPG, RAM, disk size, … you need. Microsoft provide predetermined options (more than 700).

Scaling Azure VMs

You can increase the size of a VM easily (scale up) OR yu can add more VMs and have them work together to handle the work (scale out).

Note: Scale up has some limitations (e.g. maximum CPU). Thus, preferred method is scale out.

One approach to do this is “Virtual Machine Scale Sets”. Here a group of virtual machines that can grow and shrink in quantity based one a predefined rule (based one monitoring demand / time (schedule) / other factors).

VM Scale Sets:

- Elasticity
- Two or more VM running the exact same code
- With a “load balancer” (traffic cop) in front to direct traffic randomly to one of the machines
- Able to add more machines as demand grows (autoscaling)
- Able to add more machines as demand slows
- Can handle up to 100 VMs in a single scale set
- If need, you can create more scale sets

Two main concepts when running multiple VMs are: Availability Sets and Proximity Groups

Availability Sets:

You have multiple VM that have an identical function.

You want to signify to Azure that these machines are critical and they should be kept separated each other. The reason for this is “fault isolation”. In this way if something goes wrong, it may affect one of the VM.

Fault isolation is done using:

- Fault domain: This is some type of unplanned outages (power / network) which would affect entire groups / rack of computers and then bring down all machine run on that rack. So, we do not want all the VMs run on that rack.

- Update domain: These are planned outages. Example: MS roll out new version of planform, and they do not do it to all the servers at once (different update domains). Thus, we can put our VMs in different update domains.

- Separated power sources and network switches

- Updated one at a time, not all together

The opposite of availability set is a Proximity Group. Here,

- Multiple VMs that have identical function

- Arranged in a VM Scale Set

- You want them placed together

- Less availability for more performance

- Fastest inter-server communications

Now we are moving cloud native version / things run on cloud (Azure)

App Services (Web Apps)

A new paradigm for running code in the cloud

Give your code and configurations to Azure, and they will run it.

Promise of performance but no access to hardware

Platform as a Service (PaaS)

This gives developers more benefits. It is a lot easier to develop and test code with integration (GitHub). The drawback is we do not complete access to the server. We can't install all the software we need, but only the ones supported by Microsoft.

Container Services

Another paradigm for running code in the cloud.

Here we are packaging all the files, including libraries you need to run in a “container image”. Once you built this image then you can deploy it where you want it to go (staging environment / development environment / production environment) without having recompiling it again.

MS Azure supports containers everywhere. 3 main containers:

- Azure Container Instance (ACI): Single instance, not much scaling options, quick way to deploy a container

- Azure Container Apps: Easy to use like a web service, with advanced features

- Azure Kubernetes Service (AKS): Runs on cluster of servers, enterprise-grade. Own tools, own commands

Azure Virtual Desktop

Desktop version of Windows that runs in the cloud (You will be logging with username & password, and you will have a virtual desktop)

Your software installed and your files available (from anywhere)

Can even see your desktop on IOS and Android (can even use phone) / web browser

Runs on Azure

Azure Functions

Typically, these are small pieces of code that run entirely in the cloud.

It is a utility functions that does something specific in a finite amount of time. These are not websites / long running programs. Essentially these does not worth developing whole VM for the purpose.

Is trigger by soothing happening (HTTP call, timer, message queue, ...)

Cheaper way of running code (only runs when it is needed)

Free tier – 1 million executions per month free

Can also support more complicated designs

- Durable functions
- Long-running functions
- Premium or dedicated hosting options

Examples:

- A small piece of code that runs every day at 12 am and summarizes yesterday's data.
- A small piece of code that checks a blob container for new files and does something every time it finds a new one.
- A small piece of code that runs every 6 hours and retrieve that latest weather forecast from a publicly accessible website.

Azure Networking Services (in Azure)

Microsoft call them Virtual Networking (VNets).

Microsoft has a physical network connecting all the data centers and other end points. However, here we are looking into virtual networks.

By default, two virtual machines in Azure are not allowed to talk with each other (for security reasons). By having a virtual network, then we can set up connectivity between VMs. It is an analogue to actual network of data centers.

In Azure, it is virtual because it's effectively just a database entry in a table that establishes the path between VM 1 and VM 2.

Basics of VNets:

- Virtual Networks are assigned an address space of either IPv4 and IPv6 address, or both (we define range of IP addresses)

- These are private addresses, which cannot be accessed from outside of Azure or other networks inside of Azure.

- A single VNet is usually assigned a large quantity address space to support potential future growth.

- There is no shortage or private IP addresses

VNet does not stand alone. There are things called “Subnets”.

- All VNets are subdivided into one or more subnets (minimal 1 subnet)

- The subnet is assigned a range of IP addresses which must exists in the address space of the parent VNet

- Usually there is a security layer between subnets; traffic must watch a predefined rule set to pass

Now we have VMs and subnets. How does VM working with it.

- All VM must belong to at least one subnet, using a Network Interface Card (NIC)

- Some VMs have more than one NIC and can connect to more than one subnet

- VMs can optionally be assigned a public IP, which makes it eligible to access from outside Azure (this will require to have security / firewalls)

Network Security Group (NGS)

It is an access control list (ACL) that blocks traffic inbound and outbound from a subnet unless it matches certain rules.

The rules are based on source IP, source port, destination IP, destination port, and protocol (5-tuple match).

NSGs:

- You can allow communication between different subnets in the same network through adding specific NSG rules.

- No traffic passes the NSG filter unless an “ALLOW” rule matches. (There are “DENIED” rules)

- Rule have priorities, and the highest priority rule that matches is the one that applies. (Example: If it checks “DENIED” rule with high priority, they may not allow to be passed).

Now we know how to allow communication between VM on one subnet with another VM one a different subnet. How about communication between VM an one subnet and VM on different network. We can do that using NSGs. One solution is “Peering”.

Peering: This allows communication between VM on a one network with VM on a different network (Note: make sure no conflicting IP address, that not to repeat IP addresses in different networks).

Another approach is **Azure DNS** (Domain Name System). You can give your IP addresses using a DNS. There is a private DNS in Azure, that allows you to give private addresses names (then it is easy to remember). Note: DNS only applies internally to Azure to applied networks. Examples: http://development.local, http://database.local, … We can try http://dev.mydomain.com, but this will only works within the network.

Azure VPN (Virtual Private Network) Gateway

Allows communication between a workstation and a network, or between two networks.

Encrypts traffic between those two points.

Outside of Azure, VPNs require a physical device to be installed on a network.

Inside of Azure, you can install a VPN Gateway as a virtual device on your network.

VPN Gateway requires its own subnet.

Working from home:

- When working from home, you might need to use a VPN to connect to the office network. This is called “point to site” VPN or P2S. Your computer is the “point” and company’s network is the “site”.

You can also connect two distant networks using VPN devices and this is called “**VPN Peering**”. This is called “site to site” VPN or S2S. Thus, we can connect two or more offices to an Azure subnet, or we can connect two offices with S2S.

If communicating into Azure at high speed is important to you, you may consider “**ExpressRoute**”. It is a private connection from your Internet Service Provider (ISP) to an Azure ISP (Azure endpoint) (that is, bypasses the public internet).

Public and Private Endpoints

When creating a resource in Azure (e.g., storage account), we need to consider how it is accessed and secured.

Three primary access options typically encounter:

1. Enable public access from networks (“Public Network Access”). By enabling we have public access (can access through internet). Public access does not mean it is not secure. We can have authentication / access keys / shared access signatures are required to use the resource. (example: house door facing the street, but need a key to open the door)

2. “Public network access scope”: Choose “Enable from selected virtual networks and IP addresses”. This is where we introduce service end points. With service end points, Azure resource is still containing public IP address but restricting which specific VMs and IP ranges can connect to it. This directly connect them and improve security and speed. Even someone with access key cannot reach.

3. Or we can select “Disable” on “Public network access” with Private End Points. This completely disable all public access of resource. Not even with access key. Cannot connect with public internet and virtual networks. Instead, we have to create private end points. That is uses private IP addresses from your own Azure virtual network (Azure private link). (Example: private road from your house to shop and no public access to that road).

These are available for other resources as well. This is important for security and sensitive data.