**Azure Services**

**WebApp**

An App Service plan defines a set of compute resources for a web app to run. These compute resources are analogous to the server farm in conventional web hosting. One or more apps can be configured to run on the same computing resources (or in the same App Service plan).

**Azure KUDU** is one of the fastest and easiest ways of deploying the web site components to the **Azure** website.

Entrypoint keyword makes the docker container executable

CMD keyword will provide the arguments to ENTRYPOINT command

**ARM Templates:**

Storage Account Provider: "Microsoft.Storage/storageAccounts"  
  
“Container type": blobServices/containers"

Availability Set type: "Microsoft.Compute/availabilitySets"

NetworkSecurityGroup: Microsoft.Network/networkSecurityGroups

Virtual Network: "Microsoft.Network/virtualNetworks"

NetworkInterface: "Microsoft.Network/networkInterfaces",

PublicIpAddress: "Microsoft.Network/publicIPAddresses",

VirtualMachine: "Microsoft.Compute/virtualMachines",

CORS:

Cross origin resource sharing is allows the resources(webapps, function app) to API management

**Virtual Machine:**

* Virtual machines are based on computer architectures and provide functionality of a physical computer.
* It is also capable of performing tasks such as running applications and programs like a separate computer.

**Virtual Machine Scale Set:**

* Azure virtual machine scale sets let you create and manage a group of identical, load balanced VMs. The number of VM instances can automatically increase or decrease in response to demand or a defined schedule.
* Scale sets provide high availability to your applications, and allow you to centrally manage, configure, and update a large number of VMs.

1. Instances
2. Scaling
   * Manual Scale
   * Custom Scale
3. Storage
4. Security
5. Size

* Virtual machine scale sets use disks to store the VM instance's operating system, applications, and data. As you create and manage a scale set, it is important to choose a disk size and configuration appropriate to the expected workload.

**AutoScale:**

Autoscale allows you to have the right amount of resources running to handle the load on your application. It allows you to add resources to handle increases in load and also save money by removing resources that are sitting idle.

**Availability Zone:**

Availability Zones is a high availability offering that protects your applications and data from datacenter failures. Availability Zones are unique physical locations within an Azure region. Each zone is made up of one or more datacenters equipped with independent power, cooling, and networking.

Unique physical locations within a region. Each zone is made up of one or more datacenters equipped with independent power, cooling, and networking.

**DNS Zone:**

* A DNS zone is used to host the DNS records for a particular domain. For example, the domain 'contoso.com' may contain a number of DNS records such as 'mail.contoso.com' (for a mail server) and 'www.contoso.com' (for a web site).
* Azure DNS allows you to host your DNS zone and manage your DNS records, and provides name servers that will respond to DNS queries from end users with the DNS records that you create.

**Availability Set:**

* An availability set is a group of virtual machines that are deployed across fault domains and update domains.
* Availability sets make sure that your application is not affected by single points of failure, like the network switch or the power unit of a rack of servers.

**Load balancer:**

* Azure load balancer is a layer 4 load balancer that distributes incoming traffic among healthy virtual machine instances.
* Load balancers uses a hash-based distribution algorithm. By default, it uses a 5-tuple (source IP, source port, destination IP, destination port, protocol type) hash to map traffic to available servers.
* Load balancers can either be internet-facing where it is accessible via public IP addresses, or internal where it is only accessible from a virtual network.
* Azure load balancers also support Network Address Translation (NAT) to route traffic between public and private IP addresses.

1. Frontend IP Configuration
2. Backend Pools
3. Health Probes
4. Load balancing rules
5. Inbound NAT rules

You can configure the load balancer to:

* Load balancer incoming traffic across your virtual machines.
* Forward traffic to and from a specific virtual machine using NAT rules.

**Traffic Manager:**

* Azure Traffic Manager helps reduce downtime and improve responsiveness of important applications by routing incoming traffic across multiple deployments in different regions.
* Built-in health checks and automatic re-routing help ensure high availability if a service fails.
* Use Traffic Manager with Azure services including Web Apps, Cloud Services and Virtual Machines - or combine it with on-premises services for hybrid deployments and smooth cloud migration.

Use Traffic Manager to:

* Improve app availability with automatic failover.
* Increase your app's responsiveness by routing end users to the Azure location with lowest network latency.
* Seamlessly combine on-premises and cloud.

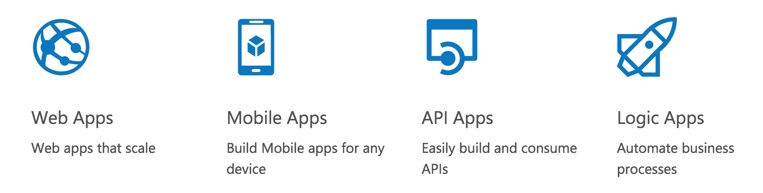
**FunctionApp:**

* Azure functions are a running small pieces of code in cloud, without worrying about a whole application or infrastructure to run it.
* It will support the multiple languages are C#(.Net or .Net core), java, JavaScript(Nodejs), python, PowerShell etc.
* Trigger(Blob storage)🡪code(function)🡪output
* Azure Functions is a serverless compute service that enables you to run code on-demand without having to explicitly provision or manage infrastructure.
* Use Azure Functions to run a script or piece of code in response to a variety of events.

**App Service Environment:**

The App Service Environment is a deployment of the Azure App Service into your own Azure Virtual Network. This enables your apps to have direct access to corporate resources over Site-to-site or ExpressRoute connections. Apps deployed in an App Service Environment operate in Isolated pricing plans and have higher scale options than are otherwise available.

* Azure App Service is a fully managed "Platform as a Service" (PaaS) that integrates Microsoft Azure Websites, Mobile Services, and BizTalk Services into a single service, adding new capabilities that enable integration with on-premises or cloud systems. Azure App Service gives users several capabilities .



1. Provision and deploy Web and Mobile Apps in seconds
2. Build engaging iOS, Android, and Windows apps
3. Automate business processes with a visual design experience
4. Integrate with "Software as a Service" (SaaS) applications (Office 365, Salesforce, Dynamics, OneDrive, Box, Dropbox, Twilio, Twitter, Facebook, Marketo, and so on) and on-premises applications

**App Service Plan:**

* In App Service, an app runs in an App Service plan. An App Service plan defines a set of compute(CPU, memory, storage, Scalability, Traffic manager) resources for a web app to run.
* These compute resources are analogous to the server farm in conventional web hosting. One or more apps can be configured to run on the same computing resources (or in the same App Service plan).

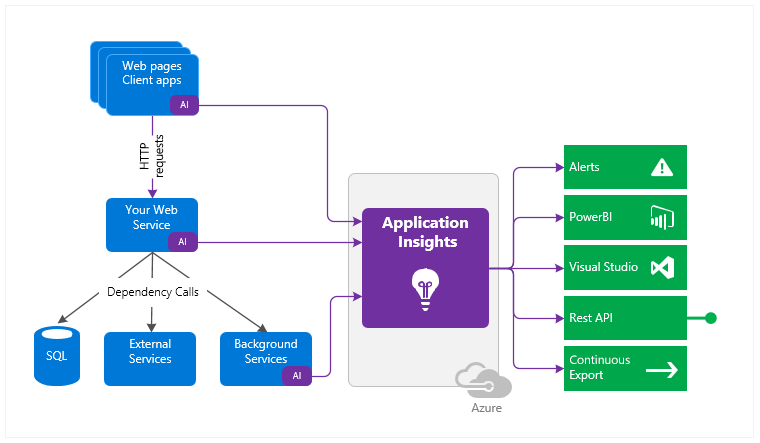
**API(Application Program Interface):**

An application program interface (API) is a set of routines, protocols, and tools for building software applications. Basically, an API specifies how software components should interact. Additionally, APIs are used when programming graphical user interface (GUI) components. A good API makes it easier to develop a program by providing all the building blocks. A programmer then puts the blocks together.

When you use an application on your mobile phone, the application connects to the Internet and sends data to a server. The server then retrieves that data, interprets it, performs the necessary actions and sends it back to your phone. The application then interprets that data and presents you with the information you wanted in a readable way. This is what an API is - all of this happens via API.

**Application Insights:**

* Application Insights is an extensible Application Performance Management (APM) service for web developers on multiple platforms.
* Use it to monitor your live web application. It will automatically detect performance anomalies.
* It includes powerful analytics tools to help you diagnose issues and to understand what users actually do with your app.
* It's designed to help you continuously improve performance and usability.



**Vnet(Virtual Network):**

* Azure Virtual Network (VNet) is the fundamental building block for your private network in Azure.
* VNet enables many types of Azure resources, such as Azure Virtual Machines (VM), to securely communicate with each other, the internet, and on-premises networks.
* VNet is similar to a traditional network that you'd operate in your own data center, but brings with it additional benefits of Azure's infrastructure such as scale, availability, and isolation.

**Vnet Peering:**

* VNet peering - connecting VNets within the same Azure region
* Global VNet peering - connecting VNets across Azure regions

The benefits of using virtual network peering, whether local or global, include:

* Network traffic between peered virtual networks is private. Traffic between the virtual networks is kept on the Microsoft backbone network. No public Internet, gateways, or encryption is required in the communication between the virtual networks.
* The ability for resources in one virtual network to communicate with resources in a different virtual network, once the virtual networks are peered.
* The ability to transfer data across Azure subscriptions, deployment models, and across Azure regions.

**Horizontal Scaling:**

Horizontal scaling means that you scale by adding more machines into your pool of resources.

**Vertical Scaling:**

Vertical scaling means that you scale by adding more power (CPU, RAM) to an existing machine

**Gateway Transit:**

Gateway Transit enables you to use a peered virtual network's gateway instead of creating a new gateway for connectivity. As you increase your workloads in Azure, you need to scale your networks across regions and virtual networks to keep up with the growth

**ARM**

Templates are idempotent, which means you can deploy the same template many times and get the same resource types in the same state.

If you need a way of deploying infrastructure-as-code to Azure, then Azure Resource Manager (ARM) Templates are the obvious way of doing it simply and repeatedly. They define the objects you want, their types, names and properties in a JSON file which can be understood by the ARM API.

The main benefits of the ARM API are that you can deploy several resources together in a single unit and that the deployments are idempotent, in that the user declares the type of resource, what name to use and which properties it should have; the ARM API will then either create a new object that matches those details or change an existing object which has the same name and type to have the same properties.

**High Avaliability and Disaster Recovery**

Fundamentally, High Availability and Disaster Recovery are aimed at the same problem: keeping systems up and running in an operational state, with the main difference being that HA is intended to handle problems while a system is running while DR is intended to handle problems after a system fails. Regardless though of how highly available a system is, any production application no matter how trivial minimally needs to have some sort of disaster recovery plan in place.

Disaster recovery picks up where high availability fails. Disaster recovery can be as simple as restoring from a backup, but it can also be very complex too depending on two factors: the Recovery Time Objective (RTO) and Recovery Point Objective (RPO).

A Recovery Time Objective is the maximum amount of time that a system can be down before it is recovered to an operational state. For some systems, this recovery time objective can be measured in hours or even days, but for more mission-critical systems the recovery time objectives are typically measured in seconds.

A Recovery Point Objective is the amount of data loss, measured in time, that is tolerable in a disaster. For some systems, losing a day’s worth of data might be acceptable while for other systems this might be mere minutes or seconds. The length of RTO’s and RPO’s have profound implications on how disaster recovery plans are implemented.