

about hardware, licenses, OS management, backup etc. and just to need to pay what they use. This is nothing but the cloud computing and why cloud came into existence.

## 2. What are the benefits, advantages of cloud computing?

- Scalability
- Agility
- High Availability
- Pay as you go
- Moving from Capex to Opex
- Fault Tolerance
- High Response Time
- High Bandwidth
- Low Latency

### Deep dive:

#### Scalability

It can be termed as an ability of the application to handle growing amount of work without degrading the performance. Basically system can grow without limits based on demand. Scalability is of 2 types –

1. Vertical or Scale Up – In this type of scaling configuration we increase the capacity of the existing hardware or machine. For example, if you are having a desktop of 4GB RAM and tomorrow you increase the RAM of your machine to 16GB then this is nothing but the vertical scaling.
2. Horizontal or Scale Out – In this type of scaling configuration we increase the count of machines without increasing the capacity. For example, if you have a server of 8GB RAM and tomorrow you put another server of same RAM and so on. When you do horizontal scaling essentially you increase the computing power in parallel so you get better performance. This is horizontal scaling.

#### Agility

The system is easy to evolve as business changes. This defines a degree to which your application can be evolved/ changed/ modified depending on the business or requirement changes.

#### High Availability

Basically this defines how much your application is available for its end users. Customers or users will always expect zero downtime of the application. High availability means your application can withstand below common types of downtime scenarios –

- Hardware failures
- Application updates
- Configuration changes
- OS update restarts
- Heavy load restarts

### *Pay as you go*

This is the most lucrative benefit of cloud. With cloud you pay only for what you use. If you are not in the need of resources, you can free them/ de-allocate them and you are not charged for it. The billing is absolutely "per minute" billing giving you more saving on cost.

### *Moving from Capex to Opex*

This stands for moving from "Capital Expenditure to Operational Expenditure". Companies today don't wish to invest on hardware as hardware is changing at rapid pace. Therefore, instead of investing on hardware cost which may become obsolete after certain time span they are more interested and willing to pay operational cost on monthly, yearly basis. This is nothing but the Capex to Opex. As Microsoft Azure being "Pay as you go" model it becomes easy to move from Capex to Opex.

### *Fault Tolerance*

It is the ability of the system to continue operating in full capacity and fully functional in the event of failure of some of its components. This simply means, if I have a web application and interacting with database then if the database is not accessible/down still my web application will be available for users. This means your application is fault tolerant.

Microsoft Azure applies various replication and redundancy strategies to make azure hosted services and applications as fault tolerant.

## 3. What is Microsoft Azure

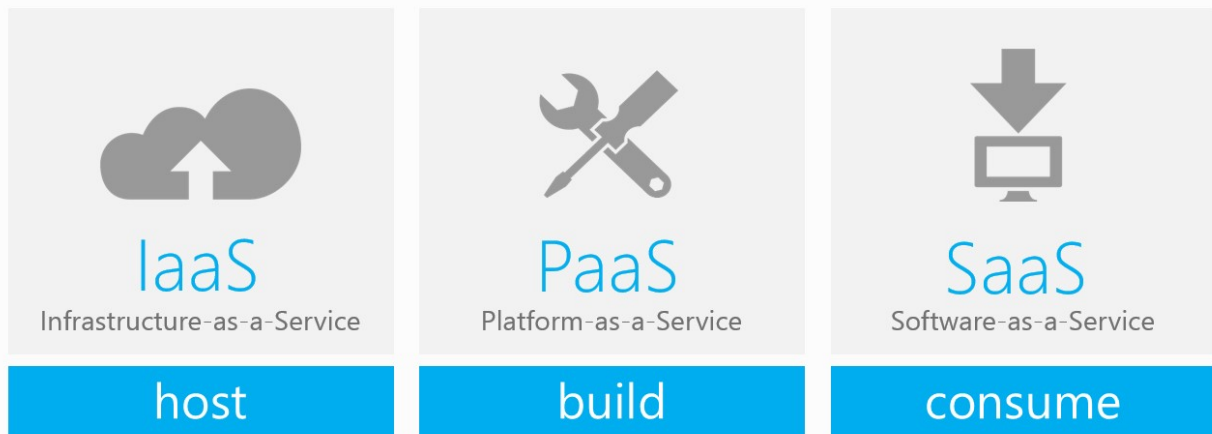
Cloud computing is the concept and this concept has been implemented by many companies. For example, Amazon implemented it and named their product as Amazon EC2 or Amazon web services, Google call it as Google App Engine and so on.

On similar lines Microsoft also created a product based on cloud computing concept and it is called as "Microsoft Azure". So MS Azure provides a platform with cloud computing capabilities so that you can get all the benefits of cloud computing.

### **Deep Dive:**

Here while answering this you can also start with answer to question "what is cloud computing" and then explain above paragraph.

#### 4. What is IaaS, PaaS and SaaS?



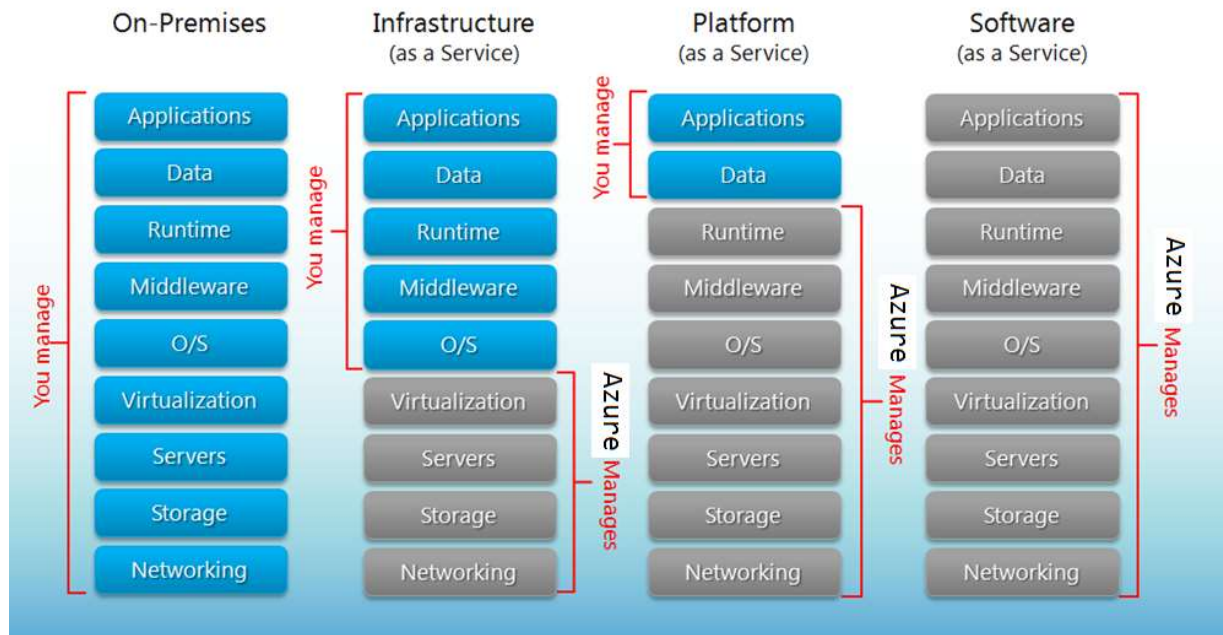
The summary of these terminologies is as follows –

1. **IaaS – Infrastructure as a Service** – a set of infrastructure level capabilities such as an operating system, network connectivity, etc. that are delivered as pay for use services and can be used to **Host** applications.  
Example, Azure VM, VNET.
2. **PaaS – Platform as a Service** – is about abstracting developers from the underlying infrastructure to enable applications to quickly be composed. This is specifically for developers who are willing to **Build** applications without worrying about management of hosting environment at all.  
Example, Azure Cloud services, Azure Web Apps, Storage, SQL Azure Database and so on.
3. **SaaS – Software as a Service** – applications that are delivered using a service delivery model where organizations can simply **Consume** and use the application. Typically, an organization would pay for the use of the application or the application could be monetized through ad revenue.  
Example, Office 365, Gmail, Salesforce.com, SharePoint online, CRM online and so on.

#### Deep Dive:

Following diagram explains important difference between IaaS, PaaS and SaaS.

# Separation of Responsibilities



As you can see from the above diagram, more you go towards right hand side of the graph things that you manage become lesser and lesser. In SaaS typically you don't manage anything and just consume. At the max you can make look and feel or configuration changes but everything rest is managed by Microsoft Azure for you.

## 5. What is Public, Private and Hybrid cloud implementation with respect to Azure?

The short answer for this questions is as below –

**Public Cloud** – All components of your application/ system are running in Azure only.

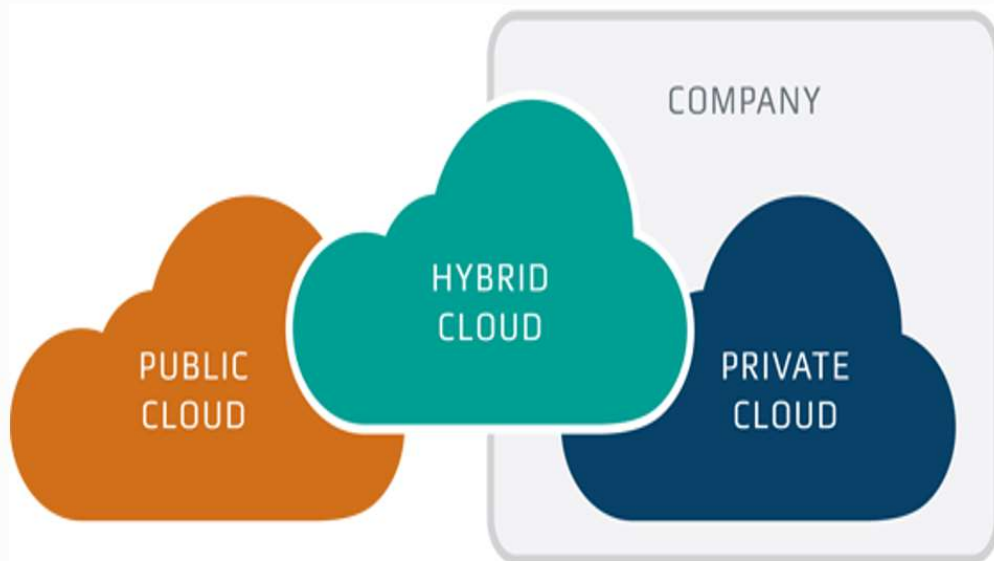
**Private Cloud** – You are running Azure services and features within on-premises data center OR you are using on premises data center for hosting your system or applications.

**Hybrid Cloud** – Combination of Public and Private. Some part or components of your application running on Azure where as some part of your application is running within on premises datacenter.

### Deep Dive:

Following diagram represents the essence of Public, Private and Hybrid cloud.

# Azure Cloud Models...



## **Public Cloud example –**

You are running an application of Hospital Management system on Azure Web Apps or Cloud service web roles and supporting backend database you are using is SQL Azure Database. In this example as all components of entire system [web app and database] are running on Azure itself this become your public cloud implementation.

## **Private Cloud example -**

You are running an application of Hospital Management system on VM in your data center within your organization premises and supporting backend database you are using is SQL server installed on VM in on premises data center. In this example as all components of entire system [web app and database] are running on premises itself this become your private cloud implementation.

Similarly, if you use Azure Stack feature, you get all of the azure service running in your private within premises data center itself. So on this data center if you run your hospital management system it will also be termed as Private cloud.

## **Hybrid Cloud example -**

You are running an application of Hospital Management system on Azure Web Apps or Cloud service web roles and supporting backend database you are using is SQL server installed on VM present within on premises data center. In this example as some of the components are running on Azure and few are running on premises this becomes Hybrid cloud implementation.

## Azure Services - Compute

This section highlights common questions around important azure services -

1. Cloud services
2. Azure Web Apps
3. Azure Virtual Machines
4. Difference between Cloud Services and Web Apps

### 6. What is Azure cloud service?

Azure cloud service is an offering from Azure and specifically designed for hosting web applications, background processing applications [similar to traditional windows service applications] and Azure IaaS workloads means Virtual Machines. Cloud Service in Azure is a container under which applications run. The web application in cloud service is termed as "Web Role" whereas background processing applications are termed as "Worker Role".

Every cloud service created from Azure provides you a DNS name such as "YourCloudServiceName.cloudapp.net". This DNS name is used for accessing web applications over internet. An Azure subscription can contain any number cloud services and one cloud service can contain any number of roles. Each role can run any number of "Instances (virtual machines)" on which actual application executes.

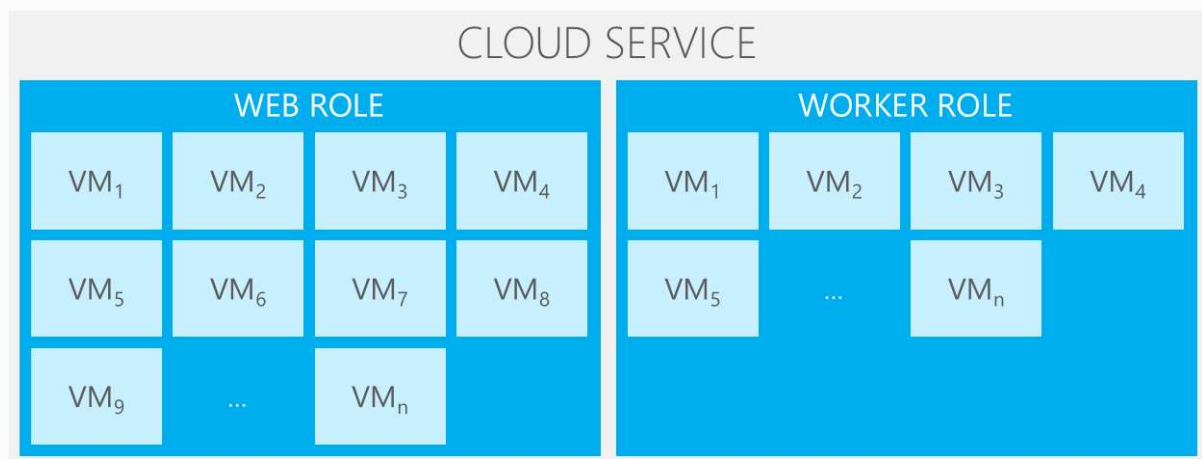
To develop an application cloud service, roles aware we need Azure SDK. Using Azure SDK one can create cloud service type of project from tools such as Visual Studio (for .net) or Eclipse (for java).

#### Deep Dive:

Following diagram shows the overall hierarchy of cloud services.

## Roles and Instances

Example Cloud Service configuration with a single web role and a single worker role



Important - Cloud services can have roles or standalone VMs running within them. When we deploy web/worker roles in cloud service it becomes PaaS offering and when we create virtual machines (by using management portal, PowerShell, CLI, Rest API or Azure SDK) under a cloud service becomes IaaS offerings.

### 7. What is purpose of cloud service configuration file (.cscfg)?

Every cloud service type of project contains .cscfg file and primarily is used for storing –

- Number of role instances to deploy for each role in cloud service project
- Thumbprint of certificates used if any
- And most important, User defined configuration settings

The primary aim or purpose of this file is to allow configuration changes in production environment without downtime of your application.

#### Deep Dive:

When you make changes to web configuration file while web application is running and active in IIS, the App Pool gets restarted, current user sessions are lost. For instance, imagine web application hosted in IIS as "desktop application". So if you save web.config file it's exactly similar to closing the opening the desktop application again. Therefore, let's say if we store azure storage connection string in web config file and after certain days or weeks you regenerated the primary key of storage account for security reasons, then you need to change it in web config file. However, if you do that, app pool will be restarted and application downtime may be experienced. Therefore, you may want to store these changing configuration settings (like azure storage connection string) in a file which is external to application binaries and deployment package so that change in it will not affect the running application at all. This can be treated as [External Configuration Store](#) Design Pattern and cloud service configuration file is the implementation of this pattern.

The .cscfg file settings can be modified from the azure portal itself. This means you don't have to redeploy the entire application which could have been the case if you have had used web.config file and needed a change in it. This is the benefit of cloud configuration file over web.config file.

As cloud configuration file is not bound with application no app pool restart scenario happens upon changes.

### 8. How to achieve zero downtime in cloud service deployments during upgrades and all hardware failures?

Run at least 2 instances of each role within a cloud service. When we have 2 or more instances running for a role in cloud service then automatically the deployment gets distributed across different fault and upgrade domains and achieves almost or near to zero downtime.

#### Deep Dive:

**Fault domain** is a physical unit of failure. In simple form when your computer is connected to power supply and if power supply is down then your computer can not be operational. So computer itself connected to power supply is a Fault domain.

**Definition** - So a fault domain is set of hardware components like computers, switches that share single point of failure.

**What is single point of failure** – is a part of the system, if it fails, will stop the entire system from working.

The infrastructure of each Azure data center is divided into multiple sections which are treated as fault domains. These sections of the infrastructure (which are not necessarily individual servers or server racks) are designed in such a way that a failure of one fault domain is extremely unlikely to affect any other fault domain. When you deploy a service, the Azure Fabric Controller automatically locates the roles in at least two different fault domains so that a failure in one domain will not affect all instances of your service.

This is the reason why Azure recommends you to have 2 instances so that Azure fabric will place your application role instances in more than one fault domain to avoid failure.

#### Upgrade Domain -

It is a logical unit of grouping the role instances and it does not exist physically. When we have 2 instances of role running then automatically each of the deployment or instances are treated as different upgrade domain. So when an upgrade of guest OS, host OS or application update need to be performed then only one upgrade domain based instances is updated while upgrade domain 2 based instance keep serving the user's requests. Once upgrade of first instance is completed then second upgrade domain based instance gets updated and so on. So at any point of time during upgrade at least one instance is alive and serving the users request hence you don't face the downtime of your application.

### 9. Why Azure recommends to run at least 2 instances of cloud service roles?

Answered as part of Deep Dive section of question no 8.

### 10. What is difference between Cloud Service "Role Instance VM" and "Cloud Service VM"?

Following table highlights important points about the difference between "Role Instance VM" and "Cloud Service VM"

Sr. No.	Role Instance VM	Cloud Service VM
1	This is PaaS offering	This is IaaS offering
2	Changes made to VM settings, data storage on drives etc. are not persistent.	Changes made to VM settings, data storage on drives etc. are <b>persistent</b> .
3	VMs provisioned for Role instances have IIS, .Net framework etc. standard pre-requisites pre-configured.	VMs provisioned have only OS with no addition standard application configured and it is user's responsibility.



4	Vertical scalability is not supported for role instance VMs. RE-deployment of entire application is required for achieving vertical scalability.	VMs can be scaled vertically without any re-deployment or re-creating the VMs.
5	Horizontal scalability configuration is easy and no need to create pre-configured VM in advance.	Horizontal scalability configuration is driven by no. of pre-configured VMs.
6	Being PaaS based offering, very limited control over underlying VMs.	Being IaaS based offering, user has full control over VMs.
7	Underlying VMs OS disks are not accessible.	Underlying VMs OS, data disks are stored in Azure Blob storage and are accessible for user.
8	Being PaaS and less dependency on VM, there is no need to perform backup of these VMs.	Being IaaS, management is entirely user's responsibility. So user has to take regular backups of underlying OS and data disks.
9	Pre-configured software templates can't be used and OS option is only limited Windows OS based versions.	Pre-configured software template images based VMs can be provisioned supporting various flavors of Linux and Windows OS versions.
10	The use of VMs is limited to only one purpose – running and hosting web applications and background jobs through worker roles.	Not limited to only running web applications.

### Deep Dive:

1. PaaS means you focus only on Application and Data. Rest is taken care by cloud computing platform. In case role instance VMs user only deploys its application and rest is managed by underlying cloud platform.  
IaaS means you get support only virtualization and after everything is your responsibility.
2. Changes made to VM are not persistent. For example, if we create a text file on C drive, then if cloud service role is recycled due to hardware failure then new VM gets allocated to Role instance and earlier save Text file will not be present.  
For example, if we create a text file on C drive, then if hardware failure occurs then new VM gets allocated to automatically and earlier saved Text file will be present.
3. VM provisioned in role instances already have all pre-requisites installed and configured for running an application.  
However, in case of IaaS VM we only get VM with OS. Everything after that is user's responsibility. So for running web application in IaaS .NET framework installation, IIS installation etc. will be required to be done by user.
4. Vertical scalability means increasing the capacity of existing VM. For example, if you are running your web application on A2 sized VM then to make application run on A4 size, you will need to redeploy the application; which will create entirely new VM. The size of role instance VM can't be changed from portal.  
In case of IaaS VM it can be directly changed from Portal and that's it.

5. As role instances being PaaS offering and VM management is done by cloud platform, horizontal scalability is unlimited and user doesn't have to create VMs in advance to support scalability. For example, if user is running application on 2 instances and due to heavy demand if he/she wishes to run the same application on 4 instances then it is just a matter of changing the instance count on Azure portal. Automatically new VMs with IIS and .NET framework will get created as well as the application will also be deployed on it and made available for serving the user request and everything happens automatically.

On the other hand, the no. of VM to be scaled horizontally is limited by pre-configured VMs. So if user wants to scale the IaaS VM horizontally then he/she will have to create the VM with same configuration in advance and then at runtime it can scale only to no. of VMs created by user in advance. This is because VM management is entirely user's responsibility in IaaS offering.

6. AS Role instance VMs are not persistent as stated in point 2, you can't really use it for any other work whereas IaaS VM can be used to perform any task, run any standard software along with your web site like SQL server and so on.
7. In case of IaaS VMs the underlying OS disk [which may be making up C drive in VM], attached data disk [which may be making other drive than C and D] are .vhd files and those are saved in Page blob under Azure storage account. So this fact opens up the opportunities of "Image mobility". Means you can take these .vhd files on premise and provision VMs out of it or vice versa.

In case of role instance VMs underlying OS disks are managed by Azure platform and we don't have access to it at all.

8. As role instance VM management is not user's responsibility there is no need to configure backup, anti-virus, OS upgrades etc. traditional managed services tasks as they are taken care by Azure platform only.

In case of Azure IaaS VMs all types of managed services tasks are user's responsibility as you are the owner of that VM.

9. As of today role instance VMs can only have Windows OS flavors to run the web applications. The fact that Azure web roles and worker roles can run Java-based applications, there might be a requirement of having Linux OS for Java application. As of today such type of scenarios are not supported.

Today you can have any type of OS version and standard software VM provisioned from Azure Marketplace for IaaS VMs. Azure provides pre-configured template images through which all standard Microsoft stack software such as SharePoint, SQL server, Dynamics AX, CRM VMs images are readily available for consumption.

10. This point is self-explanatory.

## 11. What is the best practice for achieving the High availability of applications running on Azure VM having web tier and DB tier?

For achieving high availability in case of Azure IaaS VMs – Availability sets should be used along with at least 2 instances of VMs and implement this redundancy at every web tier and DB tier.

**Deep Dive:**

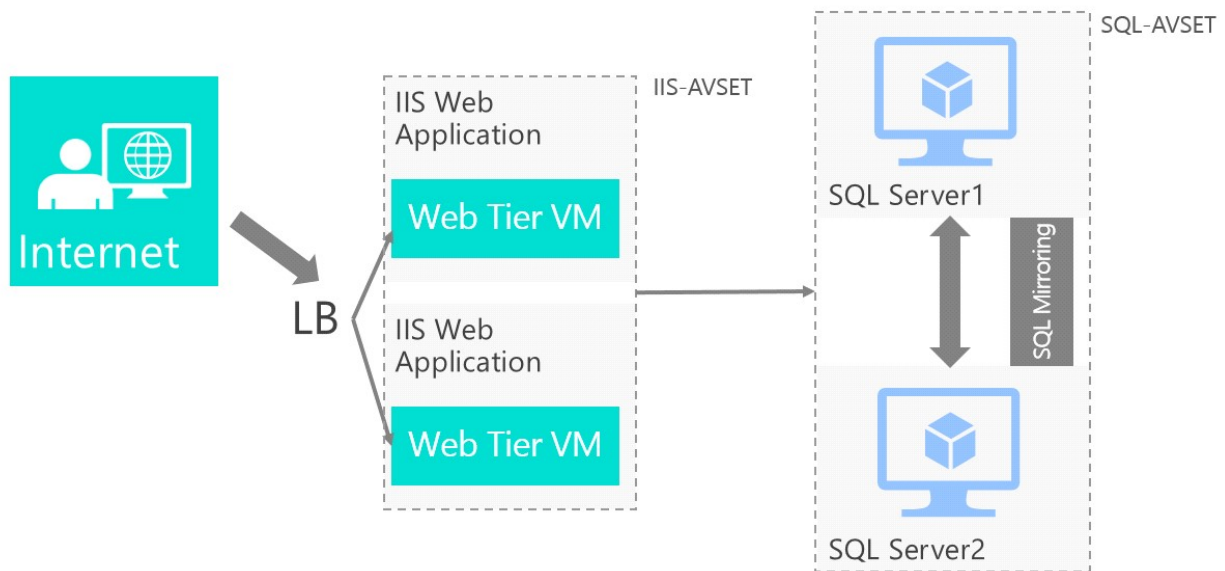
For example, let's say we want to run asp.net MVC web application along with SQL server DB on Azure IaaS VM provisioned in cloud service. Then to provide HA for VMs, it is recommended to run web application database in different server. Then have same web application hosted in 2 different VMs. Both VMs should be provisioned in the same cloud service and add them to same availability sets. Whenever Azure platform see 2 VMs part of same availability sets then automatically they are placed in different "Fault and Update domain". This way availability sets provides HA configuration for Azure VMs running web applications. On top of this it is also recommended to configure Load balancer [for the endpoint port which is used for running web application. usually this is 80 for http or 443 for https].

For database VMs again it is recommended to run them in separate Availability sets than web tier VMs. Then to keep the data in sync between 2 DB VMs either use Log shipping or mirroring mechanism. Then on top of this Internal load balancer can be used to provide communication between web tier VMs and DB tier VMs.

If cost is not the factor, then SQL Server Always On configuration is best and recommended approach for achieving HA at DB tier VMs.

Following diagram illustrates the above best practice -

## Redundancy at every level



## Azure Services - Data and Storage

This section highlights Azure storage, SQL azure specific questions.

### 12. What are different types of Azure blobs and difference between them?

Azure storage has two type of blob – Block and Page.

#	BLOCK BLOB	PAGE BLOB
1	Block blob is ideal for sequential read, write operations.	Page blob is ideal for random read, write operations.
2	Block blobs should be natural choice for storing the various types of files such as office file, pdf, mp3, video, byte array and so on.	Page blob being "sparse" type of storage, is natural choice for storing .VHD files backing the Azure Virtual Machines
3	The data constitutes from various blocks and each block can have max size of 4MB.	The data constitutes of various page ranges and each page should be in multiple of 512 bytes.
4	Write made to block blob using PutBlock is uncommitted and maintained only for 7 days. User have to call PutBlockList to commit the data permanently.	Write made to page blob using PutPage is a direct commit.

### 13. What is the difference between Table Storage and SQL Azure Table?

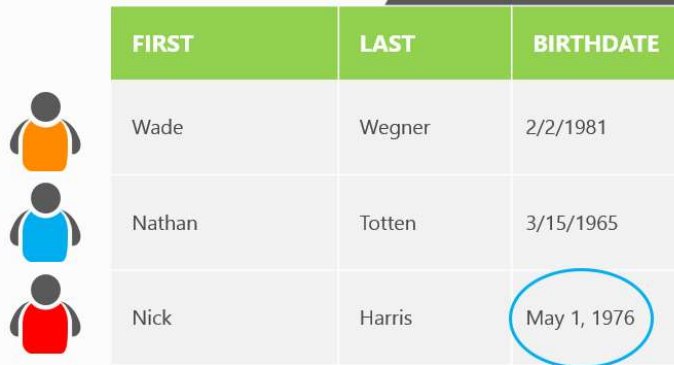
#	TABLE STORAGE	SQL AZURE TABLE
1	This is NoSQL store on Azure	This is relational store on Azure
2	As NoSQL, the data is stored in Key-Value pair combination and data is referred as an Entity.	The data is stored in Rows and Columns combination.
3	Schema is not enforced while storing the data.	Schema is enforces while storing the data. If schema is violated, then error is thrown.
4	Combination of partition and row key is treated as unique for an entity.	User can define various constraints such primary key, unique key.
5	Can't have relationship between tables.	We can define relationships between tables such as foreign key.
6	Being key-value store, we can't define objects such as Stored procedures, Views, functions.	We can created Stored procedures, views, functions.
7	General usage is observed for storing diagnostics information, error log information.	Used widely in transaction based systems.

#### Deep Dive:

##### Point no 3 -

We have seen people face difficulties in understanding the statement "Schema is not enforced on table storage". Below diagram explains this scenario well –

# No Fixed Schema



FIRST	LAST	BIRTHDATE
Wade	Wegner	2/2/1981
Nathan	Totten	3/15/1965
Nick	Harris	May 1, 1976

As you can see above the 3<sup>rd</sup> record is storing the date time in different format than first 2 records and this is absolutely accepted in Table storage. This is what we meant by "No Schema enforced on the data".

## 14. What is the difference between SQL Server on Azure VM and Azure SQL Database?

#	SQL SERVER ON VM	SQL AZURE DATABASE
1	This is IaaS offering on Azure	This is PaaS offering on Azure. It is also termed as "Database as a service (DBaaS)".
2	Access to underlying VM is available.	Access to underlying VM is not available and everything to be accessed over TDS (Tabular Data stream) based endpoint.
3	Automated backups, DR and high availability is not available and one needs to configure it.	DR, Backup and High availability is available default.
4	Eliminates Hardware cost	Eliminates hardware and administration cost as well.
5	Distributed transaction or all SQL server capabilities are supported.	<ul style="list-style-type: none"> <li>- Distributed transaction is not supported.</li> <li>- Additionally there are restrictions on the usage of some reserved keywords also.</li> <li>- Use command not supported.</li> </ul>
6	DB mirroring, Log shipping, transaction replication supported.	DB mirroring, Log shipping, transaction replication not supported.
7	SSIS, SSRS, SQL agent is available.	SSIS, SSRS, SQL agent is not available.

## Azure Networking – VNET

This section will cater to some example questions where Hybrid implementation is the key focus of Azure deployment.

15. An application front end is hosted on Azure but due to security reasons customer wants database to be hosted on-premises within his office building. What are the different ways to handle this connectivity scenario in Azure?

Looking at the requirement of connecting single on-premises DB machine to Azure hosted application, Azure VNET based "Point to Site" can be considered as correct choice in this scenario for Azure to on-premises connectivity. Point to Site is ideal choice for establishing VPN connectivity between on-premises resources and Azure resources where number of resources to be connected is limited.

16. What are the other VNET options for achieving connectivity with on-premise and Azure resources?

Site to Site and Express Route are other options for achieving cross-premises connectivity. Site to Site is specifically used when you have a large number of resources to be connected.

In some cases, Site to Site or Point to Site connectivity may introduce network latency as VPN created by these features works on public infrastructure (Internet) only. To overcome this situation "Express Route" option can be taken which offers dedicated Leased Line based offering to overcome latency issues.

17. What is the option to connect on-premises Database in case user is not willing to open up VNET based connectivity?

In such case, a WCF service can be developed and hosted on-premises. This WCF service will have CRUD operations specifically against the on-premises database. Then Service Bus relay option can be used for invoking on-premises WCF service from Azure hosted web application to access the database. Use of WCF and Service Bus relay will avoid the option of VPN connectivity using Azure VNETs offerings.

## Azure Migration

This section highlights real-world migration interview questions for IaaS or PaaS migration from on-premises to Azure.

18. On-premises application running few Windows services, console applications to handle certain tasks. What should be the approach for migration of such applications to Azure?

There are 3 ways by which we can achieve background process migration to Azure –

1. Azure Virtual Machine
2. Worker role
3. Azure Web Jobs

**Deep dive:**

The obvious way can be –

Azure VM: This is actual VM running in Azure. Current on premises console applications, windows services can be configured on this VM. Then it can function the way it was functioning on premises. However, managing the VM will be our onus.

Better way can be –

- a. Worker Role: If current windows service is written in C# [or any other Azure cloud service supported programming language] then it can be easily converted into Worker role to do the processing. Being PaaS offering we will not have to manage the role instance VM. However, in case of Worker role the scheduling part may be tricky (manual implementation) if that is the need of background application we are migrating.
- b. Web Jobs: This can run in background plus has scheduling capability as well. Webjobs runs in the context of API App service, means runs on the same instance on which web application, API app is running. Hence it saves cost as compared to Worker Role and it can scale as per the App Service plan in which web app, API app is running.

If we are looking for Micro-service based architecture (loosely couple application in all respect) –

Then, Azure functions is another way of doing it. Although it is not cost effective as webjobs but it follows “Single Responsibility Principal” providing you true nature of loosely couple applications.

## 19. What is the way to implement Caching/Session Management mechanism in Azure?

Azure Redis Cache is the most commonly used Caching mechanism available as Platform as a Service (PaaS) in Azure. In the recent announcement, Microsoft mentioned that Azure Managed Cache and In-Role cache will be retiring on 30<sup>th</sup> November 2016.

**Deep dive:**

Earlier Microsoft Azure AppFabric Distributed Cache service was the primary option for Caching implementations in Azure, however it is now deprecated and no longer recommended.

**Redis Cache:**

Redis Cache is open source Third Party service provider in Azure and it is available as Platform as a Service (PaaS). So you don't have to create VMs or configure any infrastructure components to build and utilize Redis Cache. Redis Cache is also costly Caching service compare to SQL Server option. Redis Cache have its own security layer and it is ensured to not get exposed to public networks, hence using Redis Cache for your Caching related requirements in any apps is safe. Also Redis Cache is scalable and hence very much viable solution in large enterprise applications. You can implement Redis Cache with design patterns like Circuit Breaker or Cache-Aside pattern as well.

There are three main offerings/tiers of Redis as:

1. Basic – Single instance with no SLA
2. Standard – Two instance with HA (High Availability) SLA

3. Premium – It is designed for bigger workloads and DR (disaster recovery) with all features of Standard tier and high availability SLA

You can also monitor Azure Redis Cache activities. You can also setup Redis Cache instance from PowerShell like:

```
New-AzureRmRedisCache -ResourceGroupName myGroup -Name mycache -Location "West US"
```

Based on your business requirement you can choose any tier for your application. It is highly recommended to use Redis Cache in Cloud First applications (Purely cloud based) than Hybrid Workloads/Applications.

## 20. What is Azure Resource Manager (ARM) and what are benefits of ARM over Classic services (Cloud Services)

Azure Resource Manager (ARM) is the deployment methodology/strategy to deploy your Azure components in Azure (IaaS and PaaS components). It acts like container of multiple resources however it can span across regions and services. It is template driven, declarative and idempotent in nature.

Following are the benefits of using Azure Resource Manager (ARM) over Classic Services (Cloud Services):

You can deploy, manage, and monitor all of the resources for your solution as a group, rather than handling these resources individually. You can repeatedly deploy your solution throughout the development lifecycle and have confidence your resources are deployed in a consistent state. You can use declarative templates to define your deployment. You can define the dependencies between resources so they are deployed in the correct order.

You can apply access control to all services in your resource group because Role-Based Access Control (RBAC) is natively integrated into the management platform. You can apply tags to resources to logically organize all of the resources in your subscription. You can clarify billing for your organization by viewing the rolled-up costs for the entire group or for a group of resources sharing the same tag.

## 21. What is Log Analytics (Operational Management Suite)?

Log Analytics (OMS) (formerly known by “Operational Insights”) in Azure cater all requirements in one single service and takes care of Log Analytics, Automation, Availability and Security at one single place. It provides single dashboard which gives all details of Logs, IIS Logs, Storage and other Infrastructure Log and Capacity planning details. Log Analytics also extend itself to On Premise infrastructure, Amazon (AWS) workload and Open Stack beside traditional Windows and Linux virtual infrastructure in Azure. Log Analytics also enables you to generate Power BI data source from which you can build Power BI visuals for your data. Log Analytics leverages OMS which also gives you Security and Threat Management related data with all necessary details like Login Information, IP Addresses etc. It is paid services and comes with 3 different pricing tier as Free, Standard and Premium. It helps you to search Logs across your infrastructure from a single dashboard and also allows you to export the results as well. Hence this becomes one single point/dashboard for all your logs of all your infrastructure irrespective whether it is Windows or Linux or On premise or Azure hosted.



## 22. What are the various ways by which an on premises VM can be migrated to Azure?

The best and quick way can be to migrate the associated OS disk and data disks from on premises to Azure storage, and then provision VM out of it.

For migration of on premises VM, we need to understand what is the virtualization platform used for hosting the VM. Various virtualization techniques are HyperV, VMWare etc. For creating VM on Azure requires VHD file only and that can be obtained from HyperV directly. In case VM is hosted on virtualization platform other than HyperV then we need to first convert the existing VM disks into vhd format and then can be uploaded to Azure storage. Once the disk files are present on Azure, VM can be provisioned.

## Azure Messaging – Service Bus

This section focusses on important and common questions related to Azure Service bus offering.

## 23. What is the difference between Storage queue and Service Bus queue?

Below are the important difference between storage queue and service bus queue -

SR. NO.	STORAGE QUEUE	SERVICE BUS QUEUE
1	FIFO ordering is not guaranteed	FIFO ordering of messages is guaranteed with the help of sessions
2	No session support	Messaging level sessions are supported
3	At Least Once delivery model	Supports At least once, At most once, Exactly once delivery models of messages
4	No automatic duplicate detection	Supports automatic duplicate detection
5	No support for dead-lettering	Dead letter queue is supported
6	Message size 64 KB	Message size is 256KB
7	Time to live up to 7 days max	Time to live is unlimited
8	One to one message delivery	Supports one to one and one to many message delivery
9	No transaction support	Transaction is supported
10	Batched receive is supported but batch send is not supported	Supports batched receive and batched send both.
11	Message receiving behavior is non-blocking	Message receiving behavior can be blocking or non-blocking both.

### Deep Dive:

#### Point 1 & 2 -

The storage queue does not provide guarantee about FIFO nature. The order in which messages are added to queue may not get received in the same order. On the other hand, Service Bus Queue provides FIFO ordering guarantee if "Message Sessions" are used. If messages are added with SessionID and receiver receives the messages with same SessionID then an affinity is created and the order in which messages were added, message will be received as well.

### **Point 3 – Message Delivery model –**

#### **Storage Queue - At Least Once –**

Storage Queue has two methods of receiving message from queue. "PeekMessage" and "GetMessage". Peek only read the message but it remains in the queue so other receivers can receive it again. So if you want your message to be processed multiple times then PeekMessage method should be the choice.

When GetMessage is used then received message remain invisible for some default duration for other receivers. If it is not deleted within that period, then automatically message reappears for other receivers to consume again. So typically if you want your message to be processed only once then call GetMessage method and after processing call DeleteMessage method. In this case if any error is encountered during processing of the message before calling delete message then the message will re-appear in the queue after default time period of invisibility.

As here we can see using GetMessage and DeleteMessage method combination every message in storage queue "always gets delivered or received" hence it is termed as "At Least Once" delivery.

#### **Service Bus Queue – At Most Once -**

Service bus queue message can be received using ReceiveAndDelete method. This is one way operation where message is received from service bus queue and immediately marked as complete. If message is marked for complete then automatically it gets deleted. This is one way operation hence faster and cheaper as only one transaction charges are applicable. However, if any error is encountered during processing of message after receiving then, same message is lost as and can't be received again as it has been already deleted while receive operation. So if your application can tolerate the no-processing or missing messages in some cases then ReceiveAndDelete can be used.

As the message receiving attempt is made only once and after that it gets deleted this is referred as "At Most Once".

#### **Service Bus Queue – At Least Once -**

PeekLock method of receiving message from Service Bus Queue is two stage operation. When PeekLock is called, the message gets locked and no other receivers can receive it. Once the processing is finished Complete method can be called on the same message which marks the message for delete operation. In case an error occurs while processing the message can re-appear again for receive operation after expiry of lock period. Similarly, if processing of message is required to be done immediately after error, Abandon method can be called which makes locked message available for consumption immediately. This way combination of PeekLock and Complete method makes sure that every message is received and processed always therefore this model is called as "At Least Once".

#### **Service Bus Queue – Exactly Once -**

Service bus queue supports automatic duplicate detection of message based on MessageId. If queue is created with duplication detection as true then if 2 messages with same MessageId are added to the

queue, one of them automatically will get ignored. So this way we can say Service Bus Queue support "Exactly once" delivery model if duplicate detection is enabled.

## 24. What are different receive methods can be used for Azure Service Bus Queue?

Answered as a part of deep dive section of question no. 12.

## Identity in Azure – Azure Active Directory

This section highlights important and common questions around security in Azure that are asked during interviews.

## 25. What is the difference between Azure AD and Windows Server AD?

#	WINDOWS SERVER AD	AZURE ACTIVE DIRECTORY
1	Windows AD offers 5 core services – ADCS, ADRMS, ADFS, ADDS, ADLDS.	Azure AD provides only one solution – "Identity management throughout the internet".
2	Designed to operate in corporate network where companies have full control over topology. This exposes services exposed through various protocols like LDAP, Kerberos, DNS, AD replication etc.	Type of communication is Http or Https and authentication protocols supported are OAuth, WS-Federation, SAMP-P.
3	Operations/ information can be retrieved from AD are performed through LDAP queries.	Azure AD provides Graph API which is REST API.
4	The management of devices are limited to corporate assets.	Http and https enables support for any device that can talk over http/ (s) hence enabling true Bring Your Own Device (BYOD) scenario.
5	Can't be integrated with enterprise ready SaaS applications.	The nature of Azure AD enables to integrate with enterprise grade SaaS applications such as Salesforce, Facebook at work, dropbox and so on.
6	You have access to underlying Domain controller as this is Infra offering.	No access to underlying servers as this is PaaS offering and communication happens using internet facing endpoints.