# Overview of Azure compute services

1. **Azure** **compute** is an **on**-**demand** **computing** **service** for running cloud-based applications
2. **Provides** **computing** **resources** such as **disks**, **processors**, **memory**, **networking**, and **operating** **systems**
3. You pay **only** for the **resources** you use
4. **Azure** **supports** a wide range of **computing** **solutions** for **development** and **testing**, running applications, and extending your datacentre
5. Supports **Linux**, **Windows** **Server**, **SQL** **Server**, **Oracle**, **IBM**, and **SAP**
6. Some of the most prominent services are:
   1. **Azure Virtual Machines: software** **emulations** of **physical** **computers**. In**c**lude a virtual processor, **memory**, **storage**, and **networking** resources. Create and use VMs in the cloud. **Virtual** **Machines** provides **infrastructure** **as a service (IaaS).** When you need total control over an operating system and environment, VMs are an ideal choice
   2. **Virtual machine scale sets: Used** to **deploy** and **manage** a set of **identical** **VMs**. Are designed to support true autoscale. As **demand** **goes** **up**, more **VM** **instances** can be **added**. **As** **demand** goes **down**, **VM** **instances** can be **removed**.
   3. **Containers and Kubernetes: Containers** are **lightweight**, virtualized application environments. They’re **designed** to be **quickly** **created**, **scaled** **out**, and **stopped** **dynamically**.
   4. **App Service:** you can quickly build, deploy, and scale enterprise-grade web, mobile, and API apps running on any platform. **App** **Service** is **a platform as a service (PaaS)** offering.
   5. **Functions: Ideal** when you’re **concerned** **only** about the **code** running your service and not the underlying platform or infrastructure. Used when you need to perform work in response to an event eg. REST request.

# Decide when to use Azure Virtual Machines

1. One possible solution to Tailwind Traders’ lack of physical servers is through the use of virtual machines (VMs).
2. VMs are an ideal choice when you need:
   1. **Total** **control** over the operating system (OS)
   2. The **ability** to **run** **custom** **software**
   3. To use **custom** **hosting** **configurations**
3. **Azure** **VM** **gives** you the **flexibility** of **virtualization** without having to buy and maintain the physical hardware
4. **You** still need to **configure**, **update**, and **maintain** the **software** on the **VM**.
5. Selecting an image is one of the most important decisions

## **Examples of when to use VMs**

1. **During testing and development:** Quick and easy way to create different OS and application configurations. Delete when not needed.
2. **When running applications in the cloud:**  Ability to run certain applications in the public cloud. Shutting down VMs when you don’t need them or quickly starting them up.
3. **When extending your datacenter to the cloud:** Organization can extend the capabilities of its own on-premises network.
4. **During disaster recovery:**  If a primary datacenter fails, you can create VMs running on Azure to run your critical applications

## **Move to the cloud with VMs**

1. Move from a physical server to the cloud (also known as **lift and shift**)
2. Just like a physical on-premises server, you must maintain the VM. You update the installed OS and the software it runs

## **Scale VMs in Azure**

1. You can **run** **single** **VMs** for **testing**, **development**, or **minor** **tasks**.
2. Or you can **group** **VMs** **together** to provide **high** **availability**, **scalability**, and **redundancy**.
3. Azure VM features include:
   1. **Virtual machine scale sets: Create** and **manage** a **group** of **identical**, **load**-**balanced** **VMs**. Centrally manage, configure, and update a large number of VMs in minutes. Build large-scale services for areas such as compute, big data, and container workloads.
   2. **Azure Batch:**  **Enables** **large**-**scale** **parallel** and **high**-**performance** **computing** (HPC) batch jobs. When you’re ready to run a job, Batch does the following:
      1. Starts a pool of compute VMs for you
      2. Installs applications and staging data.
      3. Runs jobs with as many tasks as you have.
      4. Identifies failures
      5. Requeues work
      6. Scales down the pool as work completes

# Decide when to use Azure App Service

1. Can deploy your application’s front-end websites to Azure App Service
2. **App** **Service** **enables** you to **build** and **host** **web** **apps**, **background** **jobs**, **mobile** **back**-**ends**, and **RESTful** **APIs** in the programming language of your choice without managing infrastructure
3. This platform as a service (**PaaS**) environment

## **Azure App Service costs**

1. You **pay** for the **Azure** **compute** **resources** your app uses while it processes requests
2. **App** **Service** **Plan** determines **how** **much** **hardware** is devoted
3. Plan determines whether it’s dedicated or shared hardware and how much memory

## **Types of app services**

1. Most common app service styles:
   1. **Web Apps: Full** **support** for hosting web apps by using ASP.NET, ASP.NET Core, Java, Ruby, Node.js, PHP, or Python in Windows or Linux
   2. **Api:**  Build **REST**-**based** web APIs by using your choice of language and framework. Get full Swagger support. Host API in Azure Marketplace. Consumed via HTTP- or HTTPS.
   3. **WebJobs:** They can be **scheduled** or run by a **trigger**. Often used to run background tasks.
   4. **Mobile Apps:** Use to quickly build a back end for **iOS** and **Android** apps. You can:
      1. **Store** **mobile** app **data** in a cloud-based SQL database
      2. **Authenticate** **customers** against common social providers, such as MSA, **Google**, **Twitter**, and **Facebook**
      3. **Send** **push** **notifications**
      4. **Execute** custom **back**-**end** **logic** in **C#** or **Node.js**
      5. On the mobile app side, there’s SDK support for native iOS and Android, Xamarin, and React native apps
2. App Service handles most of the infrastructure decisions including:
   1. **Deployment** and **management** are integrated into the platform
   2. **Endpoints** can be secured
   3. Sites can be **scaled** **quickly** to **handle** **high** **traffic**
   4. The **built-in load balancing** and traffic manager provide high availability

# Decide when to use Azure Container Instances or Azure Kubernetes Service

1. **Still** **limited** to a **single** **operating** **system** per **virtual** **machine**
2. If you want to run **multiple** **instances** of an application on a single host machine, **containers** are an excellent choice.

## **What are containers?**

1. **Containers** are a **virtualization** **environment**.
2. Run **multiple** **containers** on a **single** host
3. You **don’t** **manage** the **operating** **system** for a **container**
4. Containers are **lightweight** and designed to be created, scaled out, and stopped dynamically
5. Containers can **run side by side**
6. **Docker** the most popular container engine
7. VM virtualize the hardware
8. Containers virtualise the OS
9. With containers, **dev environment** and **prod environment** look the same

## **Manage containers**

1. Containers are managed through a container **orchestrator**
2. Start, stop, and scale out application instances as needed
3. Two ways to manage both **Docker** and **Microsoft**-**based** **containers** in Azure: **Azure** **Container** **Instances** and **Azure** **Kubernetes** **Service** **(AKS).**
4. **Azure Container Instances:** offers the fastest and simplest way to run a container in Azure. It’s a platform as a service (PaaS).
5. **Azure Kubernetes Service:** The task of automating, managing, and interacting with a large number of containers is known as orchestration. Is a complete orchestration service for containers.

## **What is Kubernetes?**

1. **Handles** **demands** of **containers** at **scale**
2. Combines **containers** and **API**
3. Manages PODS on a kubernetes cluster node
4. Minimises downtime by moving PODS/Containers went required.
5. Uses **Azure** **storage** or **Azure** **Cosmos** **DB**
6. **Kubernetes** **networking** allows access **Internet**, **load** **balancing**, **isolation** and **policy** **driven** **networks**

## **Use containers in your solutions**

1. **Containers** are often used to create solutions by using a **microservice** **architecture**
2. This **architecture** is where you **break** **solutions** into **smaller**, **independent** **pieces**
3. **For** **example**, you might **split** a **website** into a **container** **hosting** your **front** **end**, another **hosting** your **back** **end**, and a third for **storage**
4. Imagine your website back-end has reached capacity but the front end and storage aren’t being stressed. You could:
   1. Scale the back end separately to improve performance
   2. Decide to use a different storage service
   3. Replace the storage container without affecting the rest of the application

## **What is a microservice?**

1. **Microservices don’t** need to share the **same** **frameworks** or **language** etc.
2. Teams can **make** **changes** to the microservice **without** **re**-**deploying** the **entire** **application**
3. Easily **roll** **back** or **roll** **forward** an update
4. **Bug** **fixes** and new features are **less** **risky**
5. **Micorservices** must be **autonomous**
6. This provides a layer of fault isolation
7. **Microservices** **communicate** with each other **via** **API**
8. Use microservices when:
   1. High **release** velocity
   2. Highly **scalable**
   3. Rich **domains**
   4. **Small** development teams

# Decide when to use Azure Functions

1. **Azure Function:** Some of your **application** **logic** is **event** **driven**, your application is **waiting** for a particular **input** before it performs any processing. To reduce your costs, you want to avoid having to pay for the time that your application is waiting for input. With that in mind, you’ve decided to investigate Azure Functions to see if it can help.
2. **Serverless computing:** Is the **abstraction** of **servers**, **infrastructure**, and **OS**
3. Infrastructure isn’t your responsibility
4. Scaling and performance are handled automatically
5. Serverless computing includes:
   1. **Abstraction of servers:** abstracts the servers, you deploy your code, which then runs with high availability
   2. **Event-driven scale:**  Serverless computing is an excellent fit for workloads that respond to incoming events. Such as:
      1. Timers
      2. HTTP, API webhooks
      3. Queues
   3. **Micro-billing:**  Pay only for the time their code runs

# Serverless computing in Azure

1. Benefits of serverless computing in Azure:
   1. No infrastructure management.
   2. Scalability - As your application grows it can handle this
   3. Only pay for what you use
2. Azure has two implementations of serverless compute:
   1. **Azure Functions:** Functions can execute code in almost any modern language
   2. **Azure Logic Apps**: Logic apps are designed in a web-based designer and can execute logic triggered by Azure services without writing any code

## **Azure Functions**

1. When you’re **concerned** **only** about the **code**, Azure Functions is ideal
2. Used when you need to perform work in **response** to an **event**, **timer** or **message**
3. Functions **scale** **automatically**, **solid** **choice** when **demand** is **viable**
4. Using a VM you incur costs even when the virtual machine is idle
5. With **functions** you’re only **charged** for the **CPU** **time**
6. **Functions** can be either **stateless** or **stateful**
7. When they’re **stateless** (the default), they behave as if they’re **restarted** every time they respond to an event
8. When they’re **stateful** (called **Durable** **Functions**), a **context** is **passed** through the function to track prior activity
9. **Functions** are a **key** **component** of **serverless** computing

## **Azure Logic Apps**

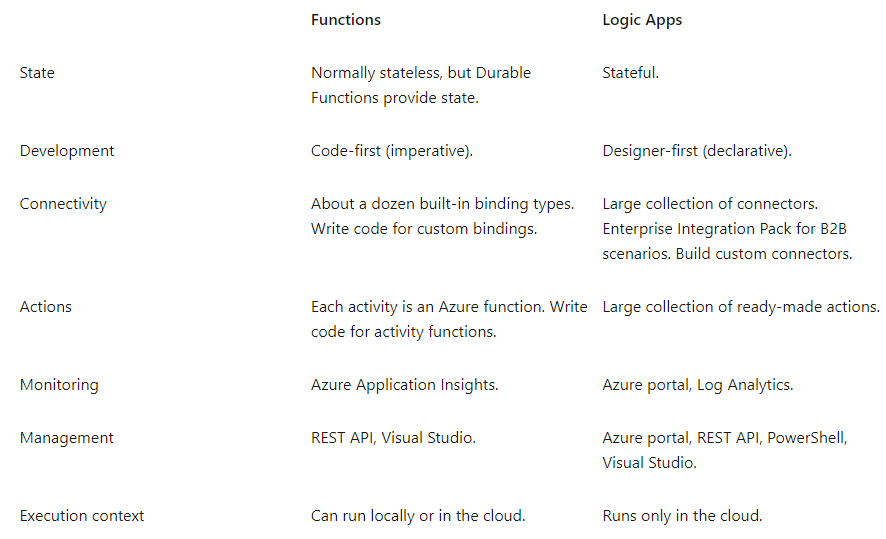
1. **Logic** **apps** are **similar** to **functions**
2. **Both** enable you to **trigger** **logic** **based** on an **event**
3. **Functions** **execute** **code**, **logic** **apps** execute **workflows**
4. Every **Azure** **logic** ap**p** workflow **starts** with a **trigger**, which fires when a specific event happens or when newly available data meets specific criteria
5. Each time the trigger fires, the Logic Apps engine creates a logic app instance that runs the actions in the workflow
6. **Actions** can also **include** **data** **conversions** and **flow** **controls**, such as **conditional** **statements**, **switch** **statements**, **loops**, and **branching**
7. You create logic app workflows by using a visual designer on the **Azure** **portal** or in **Visual** **Studio**
8. The **workflows** are persisted as a **JSON** file with a known workflow schema
9. Azure provides more than 200 different connectors and processing blocks
10. You can also build custom connectors and workflow steps
11. You then use the visual designer to link connectors and blocks together

**Example of how a Logic App would work:**

* Detect the intent of the message with cognitive services.
* Create an item in SharePoint to track the issue.
* Add the customer to your Dynamics 365 CRM system if they aren’t already in your database.
* Send a follow-up email to acknowledge their request.

## **Functions vs. Logic Apps**

1. With Functions, you write code to complete each step
2. With Logic Apps, you use a GUI to define the action
3. Here are some common differences between the two:



# Decide when to use Azure Virtual Desktop

1. With a **team** of **remote** **workers**, with **different** **OS** and **devices**. A way to **rollout** **tools** and **minimise** costs and to expedite the process Azure Virtual Desktop can be used.

## **What is Azure Virtual Desktop?**

1. Is a **desktop** and **application** **virtualization** service that **runs** on the **cloud**
2. Use a **cloud**-**hosted** **version** of **Windows** from any location
3. **Works across devices like Windows, Mac, iOS, Android, and Linux**
4. You can also use most **modern** **browsers**

## **Why should you use Azure Virtual Desktop?**

1. **Provide the best user experience:** Users have the freedom to connect to Azure Virtual Desktop with any device over the internet. They use an ***Azure Virtual Desktop client*** to connect. This could be native application or Azure Virtual Desktop HTML5 web client. User profiles are containerized by using ***FSLogix***. Individual ownership through personal (persistent) desktops. They can add or remove programs without impacting other users on that remote desktop.
2. **Enhance security:** Azure Virtual Desktop provides centralized security management. **Enable** **multifactor** **authentication** to secure user sign-ins. Can also apply granular role-based access controls (RBACs) to users. Data and apps are separated from the local hardware***. Risk of confidential data being left on a personal device is reduced***. Azure Virtual Desktop also improves security by using reverse connect technology. This connection type is more secure than the Remote Desktop Protocol. We don’t open inbound ports to the session host VMs.

## **What are some key features of Azure Virtual Desktop?**

1. **Simplified management:** You use Azure AD and RBACs to manage access to resources.
2. **Performance management:** Azure Virtual Desktop gives you options to load balance users on your VM host pools. Host pools are collections of VMs with the same configuration assigned to multiple users. For the best performance, you can configure load balancing to occur as users sign in (breadth mode). With breadth mode, users are sequentially allocated across the host pool for your workload.
3. **Multi-session Windows 10 deployment:** The only Windows client-based operating system that enables multiple concurrent users on a single VM.

## **How can you reduce costs with Azure Virtual Desktop?**

1. **Azure** **Virtual** **Desktop** is **available** to you at **no** **additional** **cost** if you have an eligible Microsoft 365 license
2. **Bring** your **eligible** **Windows** or **Microsoft** **365** **license** to get Windows 10 Enterprise and Windows 7 Enterprise desktops and apps at no additional cost.
3. If you’re an eligible Microsoft Remote Desktop Services Client Access License customer, Windows Server Remote Desktop Services desktops and apps are available at no additional cost.
4. Buy one-year or three-year Azure Reserved Virtual Machine Instances to save you up to 72 percent versus pay-as-you-go pricing

**Knowledge Check**

https://docs.microsoft.com/en-us/learn/modules/azure-compute-fundamentals/knowledge-check

Top of Form

Bottom of Form

## **Azure Virtual Network fundamentals**

## **What is Azure virtual networking?**

1. Azure virtual networks **enable** Azure resources, such as VMs, web apps, and databases, to **communicate** with each other
2. Azure network is a set of resources that links other Azure resources
3. **Azure virtual networks provide the following key networking capabilities: CCCIIFR -** Isolation and segmentation, Internet communications, Communicate between Azure resources, Communicate with on-premises resources, Route network traffic, Filter network traffic, Connect virtual networks

## **Isolation and segmentation**

* When you set up a virtual network, you define a private IP address space by using either public or private IP address ranges
* You can use the name resolution service that’s built in to Azure
* Can use either an internal or an external DNS server

## **Internet communications**

* A VM in Azure can connect to the internet by default
* Enable incoming connections from the internet by defining a public IP address or a public load balancer
* VM management via the Azure CLI, Remote Desktop Protocol, or Secure Shell

## **Communicate between Azure resources**

* Enable Azure resources to communicate securely in two ways:
  1. **Virtual networks:** Connect not only VMs but other Azure resources, such as the App Service Environment for Power Apps, Azure Kubernetes Service, and Azure virtual machine scale sets
  2. **Service endpoints:**  service endpoints to connect to other Azure resource types, such as Azure SQL databases and storage accounts

## **Communicate with on-premises resources**

* There are three mechanisms for you to achieve this connectivity:
  1. **Point-to-site virtual private networks:** Connection is from a computer outside your organization, back into your corporate network
  2. **Site-to-site virtual private networks:** A site-to-site VPN links your on-premises VPN device or gateway to the Azure VPN gateway in a virtual network
  3. **Azure ExpressRoute:** Environments where you need greater bandwidth and even higher levels of security

## **Route network traffic**

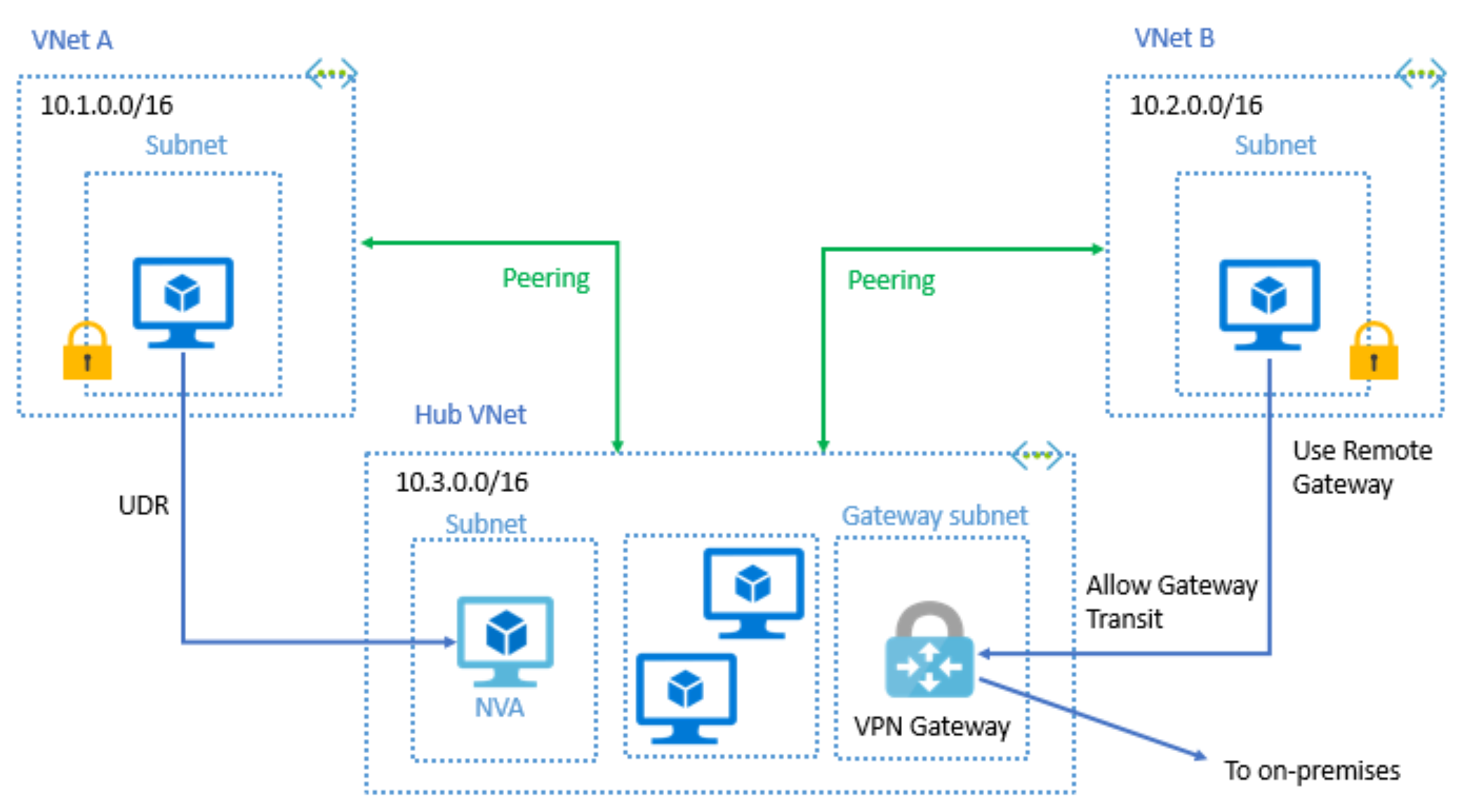
* You also can control routing and override those settings, as follows:
  1. **Route tables:** Allows you to define rules about how traffic should be directed
  2. **Border Gateway Protocol:** Works with Azure VPN gateways or ExpressRoute

## **Filter network traffic**

* Filter traffic between subnets by using the following approaches:
  1. **Network security groups:**  A network security group is an Azure resource that can contain multiple inbound and outbound security rules
  2. **Network virtual appliances:** Carries out a particular network function, such as running a firewall or performing wide area network (WAN) optimization

## **Connect virtual networks**

* **Link** **virtual** **networks** together by using virtual **network** **peering**
* Peering enables resources in each virtual network to communicate with each other
* **UDR** is User-Defined Routing
* **UDR** is a significant update to Azure’s Virtual Networks - allowing for **greater** **control** over network traffic flow.



# Azure Virtual Network Settings

1. You can create and configure Azure Virtual Network instances from the Azure portal, Azure PowerShell on your local computer, or Azure Cloud Shell.

## **Create a virtual network**

1. **Network name**: name must be unique in your subscription
2. **Address space**: define the internal address space in Classless Interdomain Routing (CIDR) format
3. **Subscription**: Only if you have multiple subscriptions
4. **Resource group:**
5. **Location** : where you want the virtual network to exist
6. **Subnet**: can create one or more subnets
7. **DDoS protection:**  Basic or Standard DDoS protection. Standard DDoS protection is a premium service.
8. **Service endpoints:** include Azure Cosmos DB, Azure Service Bus, Azure Key Vault

## **Define additional settings**

1. **Network security group:** filter the type of network traffic
2. **Route table:** Azure automatically creates a route table for each subnet

## **Configure virtual networks**

1. Make changes in Azure portal or Powershell commands

# Azure VPN Gateway fundamentals

1. **VPNs** use an **encrypted** **tunnel** within another network
2. Typically **deployed** to **connect** **two** or **more** **trusted** **private** **networks** to one another over an untrusted network (typically the public internet)
3. **Traffic** is **encrypted** **preventing** **eavesdropping** or other attacks

## **VPN gateways**

1. Azure VPN Gateway instances are deployed in Azure Virtual Network instances
2. Enable the following connectivity:
   1. Connect on-premises datacenters to virtual networks through a **site-to-site connection**
   2. Connect individual devices to virtual networks through **a point-to-site connection**
   3. Connect virtual networks to other virtual networks through a **network-to-network connection**
3. You can **deploy** **only** **one** **VPN** gateway in each **virtual** **network**
4. But you can use one gateway to connect to multiple locations
5. When you deploy a VPN gateway, you specify the **VPN** **type**: either**policy-based or route-based**
6. Difference between these two types of VPNs is how traffic to be encrypted
7. Both types of VPN gateways use a pre-shared key as the only method of authentication
8. Both types also rely on **Internet** **Key** **Exchange** (IKE) in either version 1 or version 2 and Internet Protocol Security (IPSec)

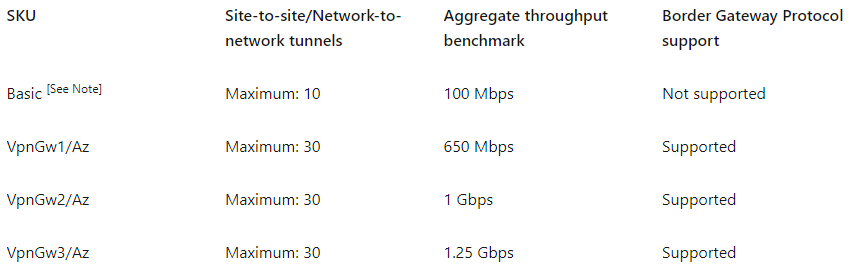
## **Policy-based VPNs**

1. **Specify** **statically** the **IP** **address** of packets that should be encrypted
2. **Device** **evaluates** **every** **data** **packet** against those sets of IP addresses
3. Key features of policy-based VPN:
   1. Support for **IKEv1** only
   2. Use of **static** **routing**
   3. Must be used in specific scenarios that require them, such as for compatibility with legacy on-premises VPN devices

## **Route-based VPNs**

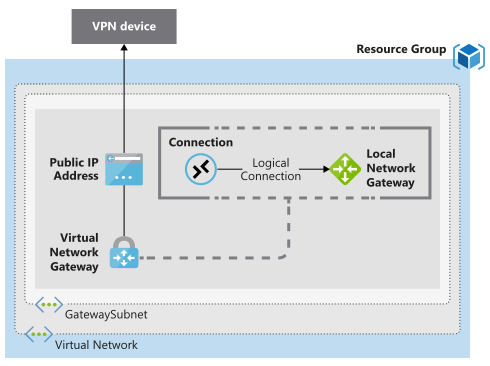
1. Used when defining which IP addresses are behind each tunnel is too **cumbersome**
2. IPSec **tunnels** are modelled as a network interface or virtual tunnel interface
3. IP routing decides which one of these tunnel interfaces
4. Are the **preferred** **connection** method for **on**-**premises** devices
5. More **resilient** to **topology** **changes**
6. Use a route-based VPN gateway when:
   1. Connections between virtual networks
   2. **Point-to-site** connections
   3. **Multisite** connections
   4. Coexistence with an Azure ExpressRoute gateway
7. Key Features:
   1. Supports **IKEv2**
   2. Uses any-to-any (wildcard) traffic selectors
   3. Can use **dynamic** **routing** **protocols**

## **VPN gateway sizes**



## **Deploy VPN gateways**

1. Azure resources you will need, before you can deploy an operational VPN gateway:
   1. **Virtual network**
   2. **GatewaySubnet**
   3. **Public IP address**
   4. **Local network gateway**
   5. **Virtual network gateway**
   6. **Connection**



## **Required on-premises resources**

1. To connect your datacenter to a VPN gateway you will need:
   1. A **VPN** **device** that supports **policy**-**based** or **route**-**based** **VPN** gateways
   2. A public-facing (internet-routable) **IPv4** **address**

## **High-availability scenarios**

1. There are several ways to ensure you have a fault-tolerant configuration:
   1. **Active/standby:** By default, VPN gateways are **deployed** as **two** **instances** in an active/standby. When planned maintenance or unplanned disruption the ***standby instance automatically assumes responsibility***. Connections are interrupted during this failover, but they’re typically restored within a few seconds.
   2. **Active/active:**  Support for the **BGP** routing protocol. You can extend the high availability by ***deploying an additional VPN device on-premises***
   3. **ExpressRoute failover:** Configure a VPN gateway as a secure ***failover path for ExpressRoute connections.*** ExpressRoute circuits have resiliency built in. Are not immune to physical problems.
   4. **Zone-redundant gateways*: In regions that support availability zones***, ***VPN gateways and ExpressRoute gateways can be deployed*** in a zone-redundant configuration. Configuration brings resiliency, scalability, and higher availability to virtual network gateways. These gateways require different gateway SKUs and use Standard public IP addresses instead of Basic public IP addresses.

# Azure ExpressRoute fundamentals

1. ExpressRoute lets ***you extend your on-premises networks*** into the Microsoft cloud
2. Establish connections to Microsoft cloud services, such as Microsoft Azure and Microsoft 365.
3. Connectivity can be from **an any-to-any (IP VPN)** network, a point-to-point Ethernet network, or a virtual cross-connection through a connectivity provider at a colocation facility
4. ExpressRoute connections ***don’t go over*** the public ***Internet***
5. Allows ExpressRoute connections to offer more reliability, faster speeds, consistent latencies, and higher security than typical connections over the Internet
6. Open Systems Interconnection (OSI) model:
   1. **Layer 2 (L2)**: This layer is the Data Link Layer
   2. **Layer 3 (L3)**: This layer is the Network Layer

## **Features and benefits of ExpressRoute**

1. Benefits to using ExpressRoute:
   1. **Layer 3 connectivity:** provides Layer 3 (address-level) connectivity. Can be point-to-point or virtual cross connections
   2. **Built-in redundancy:** Ensures that connections established with Microsoft are highly available
   3. **Connectivity to Microsoft cloud services:** Direct access to Microsoft Office 365, Microsoft Dynamics 365, Azure Compute Services and Azure Cloud Services
   4. **Across on-premises connectivity with ExpressRoute Global Reach:** To exchange data across your on-premises sites by connecting your ExpressRoute circuits. You can connect your private datacenters through two ExpressRoute.
   5. **Dynamic routing:** Uses the Border Gateway Protocol (BGP).
   6. **Connection uptime SLA.**
   7. **QoS support for Skype for business**

## **ExpressRoute connectivity models**

1. **Colocation at a cloud exchange:** ProvidesLayer 2 and Layer 3**.** If your datacenter is colocated at ***a cloud exchange such as an ISP***, you can request a virtual cross-connection to the Microsoft cloud
2. **Point-to-point Ethernet connection:** ProvidesLayer 2 and Layer 3. If you have an ***on-premises datacenter***, you can use a point-to-point Ethernet link to connect to Microsoft
3. **Any-to-any networks:** You can integrate your wide area network ***(WAN) with Azure***. offer Layer 3 connectivity

## **Security considerations**

1. With ExpressRoute, your data ***doesn't travel over the public internet***, so it's not exposed to the potential risks associated with internet communications
2. ExpressRoute is a ***private connection*** from your on-premises infrastructure to your Azure
3. DNS queries, certificate revocation list checking, and Azure Content Delivery Network requests are still sent over the public internet.

**Knowledge Check**

https://docs.microsoft.com/en-us/learn/modules/azure-networking-fundamentals/knowledge-check

# Azure Storage account fundamentals

1. [Azure Storage](https://azure.microsoft.com/product-categories/storage), which is a service that you can use to store **files**, **messages**, **tables**, and other types of information
2. Clients such as **websites**, **mobile** **apps**, **desktop** **applications**, and many other types of custom solutions can read data from and write data to **Azure** **Storage**
3. Azure Storage is also used by **infrastructure as a service** virtual machines, and **platform as a service** cloud services.
4. **Blob** stor**a**ge for **massive** **unstructured** **data**. Ideal for images or documents to browser. **Video** and **audio**.
5. To **begin** using Azure Storage, you first create an **Azure** **Storage** **Account** to store your data objects
6. You can **create** an **Azure** **Storage** account by using the **Azure** **portal**, **PowerShell**, or the **Azure** **CLI**
7. A storage account provides a unique namespace for your Azure Storage data, that's accessible from anywhere in the world over HTTP or HTTPS
8. Data in this account is secure, highly available, durable, and massively scalable
9. Check Azure Storage Video again

## **Disk storage fundamentals**

1. Disk Storage provides disks for Azure virtual machines
2. Applications and services can access these disks, as in on-premises scenarios
3. Disk Storage allows data to be **persistently** **stored** and **accessed**
4. Get solid-state drives (**SSDs**) and traditional spinning hard disk drives (HDDs),
5. Premium **SSD** disks for **mission**-**critical** production applications
6. **Standard** **SSD** and **HDD** disks for **less** **critical** workloads
7. **Ultra** **disks** for **data**-**intensive** workloads such as **SAP**

## **Azure Blob storage fundamentals**

1. Azure **Blob** **Storage** is an **object** **storage** solution for the cloud
2. It can store **massive** **amounts** of **data**, such as text or binary data
3. Azure Blob Storage is **unstructured**, meaning that there are no restrictions on the kinds of data
4. Blob Storage can manage:
   1. **Thousands** of simultaneous uploads
   2. **Massive** amounts of **video** data
   3. **Constantly** **growing** log files
   4. **Reached** from **anywhere** with an internet connection
5. Blobs aren't limited to common file formats
6. **Advantage:** Developers don’t have to think about or manage disks
7. Blob Storage is ideal for:
   1. Serving **images** or documents directly to a browser
   2. Storing **files** for **distributed** **access**
   3. Streaming **video** and **audio**
   4. Storing data for **backup** and **restore**, **disaster** **recovery**, and **archiving**
   5. Storing data for analysis by an on-premises or Azure-hosted service
   6. Storing up to **8 TB** of data

## **Azure Files fundamentals**

1. Azure Files offers fully managed file shares in the cloud
2. ***Accessible*** ***via Server Message Block*** and ***Network File System*** (preview) protocols
3. Typical usage scenarios would be to share files anywhere in the world, diagnostic data, or application data sharing
4. Use Azure Files for the following situations:
   1. Azure files makes it easier to ***migrate*** applications that share data to Azure
   2. If you ***mount*** the Azure file share to the same drive letter that the on-premises application uses, there should be minimal changes
   3. ***Store configuration files*** on a file share and access them from multiple VMs.
   4. ***Write*** ***data*** to a file share, and process or analyze the data later
5. You can also use **Shared Access Signature (SAS) tokens** to **allow** **access** to a private asset for a **specific** amount of **time**

## **Understand Blob access tiers**

1. Data stored in the cloud can be different based on how it's **generated**, **processed**, and **accessed** over its lifetime
2. To accommodate these different access needs, Azure provides several access tiers
3. You can use to balance your storage costs with your access needs
4. The available access tiers include:
   1. **Hot access tier**: data that is accessed frequently, i.e. Website
   2. **Cool access tier**:  data that is infrequently accessed and stored for at least 30 days, i.e. customer invoices
   3. **Archive access tier**:  data that is rarely accessed and stored for at least 180 days
5. **Blob access tiers considerations:**
   1. Only the hot and cool access tiers can be set at the account level. The archive access tier isn't available at the account level
   2. Hot, cool, and archive tiers can be set at the blob level
   3. Data in the cool access tier can tolerate slightly lower availability, but still requires high durability, retrieval latency, and throughput characteristics similar to hot data
   4. Archive storage offers the lowest storage costs, but also the highest costs to rehydrate and access data.

## **Knowledge Check**

https://docs.microsoft.com/en-us/learn/modules/azure-storage-fundamentals/knowledge-check

# Explore Azure Cosmos DB

1. Azure **Cosmos** **DB** is a **globally** **distributed**, multi-model database service
2. **Fast**, **single**-**digit**-**millisecond** data access by using any one of several popular APIs
3. Service level agreements for throughput, latency, availability, and consistency guarantees
4. Supports **schema**-**less** **data**
5. Azure Cosmos DB **stores** da**t**a in **atom**-**record**-**sequence** (**ARS**) format
6. This data can be abstracted and projected as an API
7. Developers can use **SQL**, **MongoDB**, Cassandra, Tables, and Gremlin

## **Explore Azure SQL Database**

1. Is a **relational** **database**
2. SQL Database is a high-performance, reliable, fully managed, and secure database
3. Platform as a service (**PaaS**) database engine
4. Handles most of the database management functions, such as upgrading, patching, backups, and monitoring, without user involvement
5. Allows relational data and non-relational structures, such as graphs, JSON, spatial, and XML

## **Migration**

1. You can migrate your existing SQL Server databases with minimal downtime by using the **Azure** **Database** **Migration** **Service**
2. Microsoft Data Migration Assistant can generate assessment reports that provide recommendations

# Explore Azure database for MySQL

1. Azure Database for **MySQL** is a **relational** **database** **service** in the cloud
2. Also has service level agreement from Azure
3. Azure Database for MySQL delivers:
   1. Built-in high **availability** with no additional cost
   2. Predictable performance and inclusive, **pay**-**as**-**you**-**go** pricing
   3. **Scale** as needed, within seconds
   4. Ability to **protect** **sensitive** data at-rest and in-motion
   5. **Automatic** **backups**
   6. Enterprise-grade security and compliance
4. These capabilities require almost no administration, and all are provided at no additional cost
5. Migrate your existing MySQL databases
6. Uses **LAMP** **Stack**

# Explore Azure Database for PostgreSQL

1. Azure Database for PostgreSQL is a **relational** **database** service in the cloud
2. Azure Database for PostgreSQL delivers the following benefits:
   1. Built-in high **availability** compared to on-premises resources
   2. **Simple** and **flexible** **pricing**
   3. **Scale** **up** or **down** as needed
   4. Adjustable **automatic** **backups**
   5. Enterprise-grade security
3. Azure Database for PostgreSQL is available in two deployment options: **Single Server** and **Hyperscale (Citus)**.

## **Single Server**

1. Focus on **rapid** **application** **development** and accelerating your time to market, rather than having to manage virtual machines and infrastructure
2. Single Server **deployment** **option** offers three pricing tiers: **Basic**, **General** **Purpose**, and **Memory** **Optimized**

## **Hyperscale (Citus)**

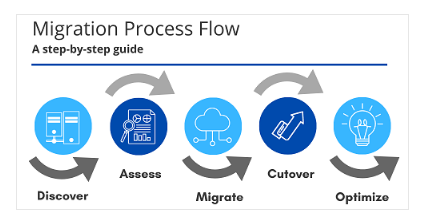
1. Horizontally scales queries across **multiple** machines by using sharding
2. Query engine parallelizes incoming SQL queries across these servers for faster responses on **large** **datasets**

# Explore Azure SQL Managed Instance

1. Azure SQL Managed Instance is a platform as a service (**PaaS**) database engine
2. You'll also be able to **protect** your **data** with **automated** **backups** and a configurable backup retention **period**
3. Azure SQL Managed Instance provides several options that might not be available to Azure SQL Database. For example DB migration with Cyrillic characters for collation

## **Migration**

1. **Azure** **SQL** **Managed** **Instance** makes it easy to migrate your on-premises data on SQL Server to the cloud using the **Azure** **Database** **Migration** **Service** (**DMS**) or native backup and restore



# Explore big data and analytics

1. Big data = large volumes of data
2. Azure supports a broad range of technologies and services:
   1. **Azure Synapse Analytics**
      1. ***Limitless*** analytics service that brings together enterprise data warehousing and big data analytics
      2. ***Query*** data using either ***serverless*** or provisioned resources at scale
   2. **Azure HDInsight**
      1. Fully managed, ***open-source analytic***s service
      2. Cloud service that makes it easier, faster, and more cost-effective to process massive amounts of data
      3. You can ***run*** popular ***open-source frameworks*** and create cluster types such as [Apache Spark](https://docs.microsoft.com/en-us/azure/hdinsight/spark/apache-spark-overview), [Apache Hadoop](https://docs.microsoft.com/en-us/azure/hdinsight/hadoop/apache-hadoop-introduction), [Apache Kafka](https://docs.microsoft.com/en-us/azure/hdinsight/kafka/apache-kafka-introduction), [Apache HBase](https://docs.microsoft.com/en-us/azure/hdinsight/hbase/apache-hbase-overview), [Apache Storm](https://docs.microsoft.com/en-us/azure/hdinsight/storm/apache-storm-overview), and [Machine Learning Services](https://docs.microsoft.com/en-us/azure/hdinsight/r-server/r-server-overview)
      4. HDInsight also supports a broad range of scenarios such as extraction, transformation, and loading (ETL), data warehousing, machine learning, and IoT
   3. **Azure Databricks**
      1. Helps you ***unlock*** ***insights*** from all your data and build artificial intelligence solutions
      2. Azure Databricks supports Python, Scala, R, Java, and SQL
   4. **Azure Data Lake Analytics**
      1. Is ***an on-demand analytics job*** service that simplifies big data
      2. Instead of deploying, configuring, and tuning hardware, you write queries to transform your data and extract valuable insights

## **Knowledge Check**

**https://docs.microsoft.com/en-us/learn/modules/azure-database-fundamentals/knowledge-check**