AZ-900

Azure Fundamentals

Getting Started

Advisor

Machine Learning

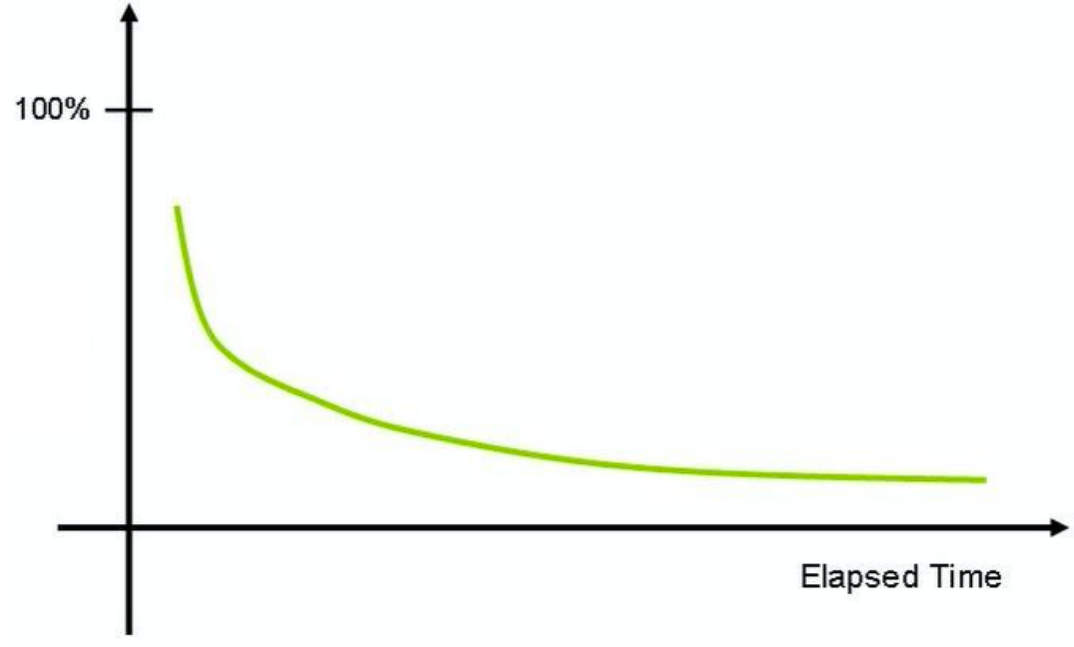
Cosmos DB

Azure DevOps

- Azure has *200+ services*. Exam expects you to understand *40+ services*.
- Exam *tests* your **decision making abilities**:
  - Which service do you choose in which situation?
- This course is **designed** to help you *make these choices*
- **Our Goal** : Help you get certified and start your cloud journey with Azure

How do you put your best foot forward?

- **Challenging certification** - Expects you to understand and **REMEMBER** a number of services
- As time passes, humans forget things.
- How do you improve your chances of remembering things?
  - **Active learning** - think and take notes
  - **Review** the presentation every once in a while

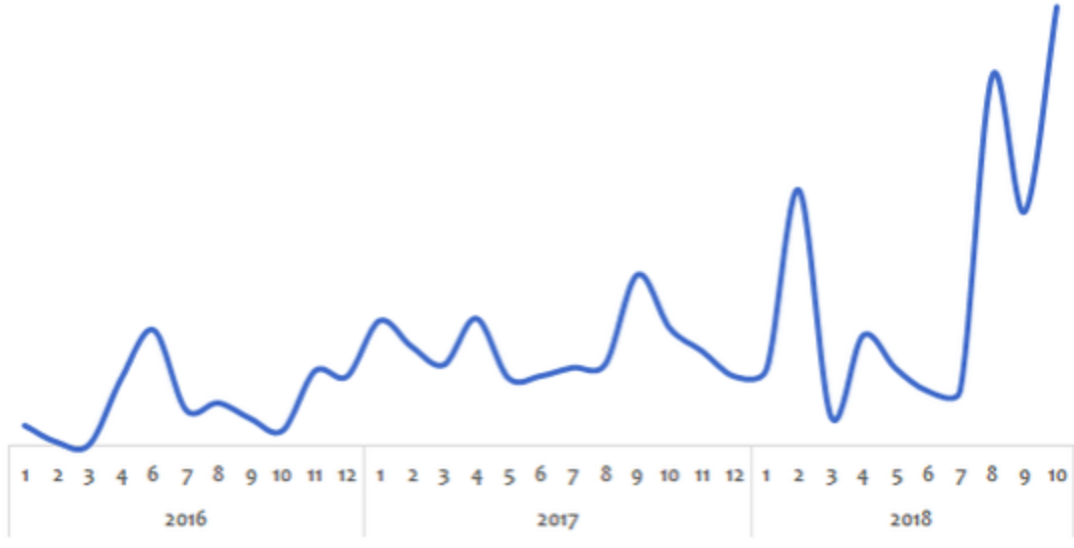


Our Approach

- Three-pronged approach to reinforce concepts:
  - Presentations (Video)
  - Demos (Video)
  - **Two kinds of quizzes**:
    - Text quizzes
    - Video quizzes
- (Recommended) Take your time. Do not hesitate to replay videos!
- (Recommended) Have Fun!

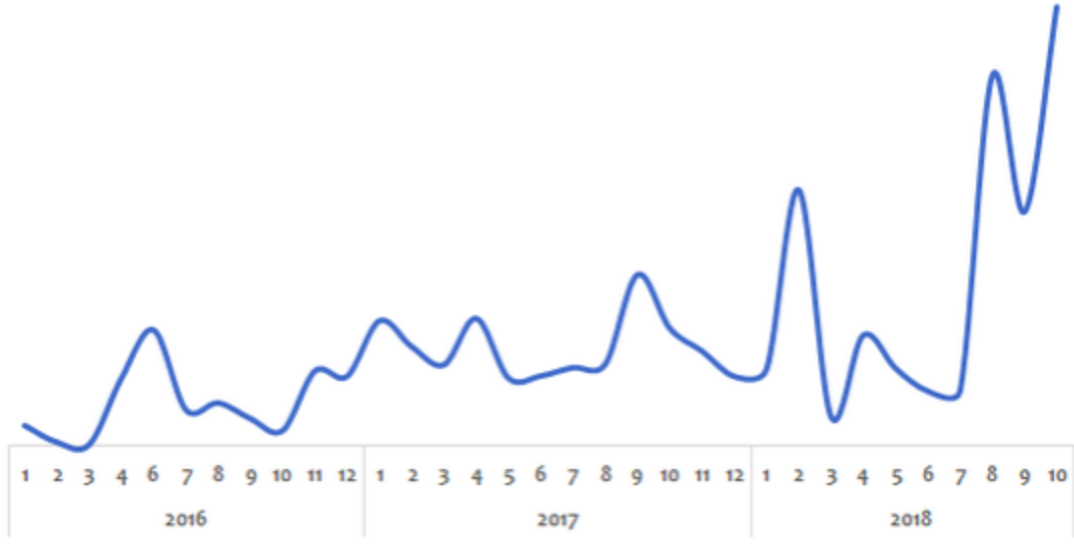
Getting Started - Azure

Before the Cloud - Example 1 - Online Shopping App



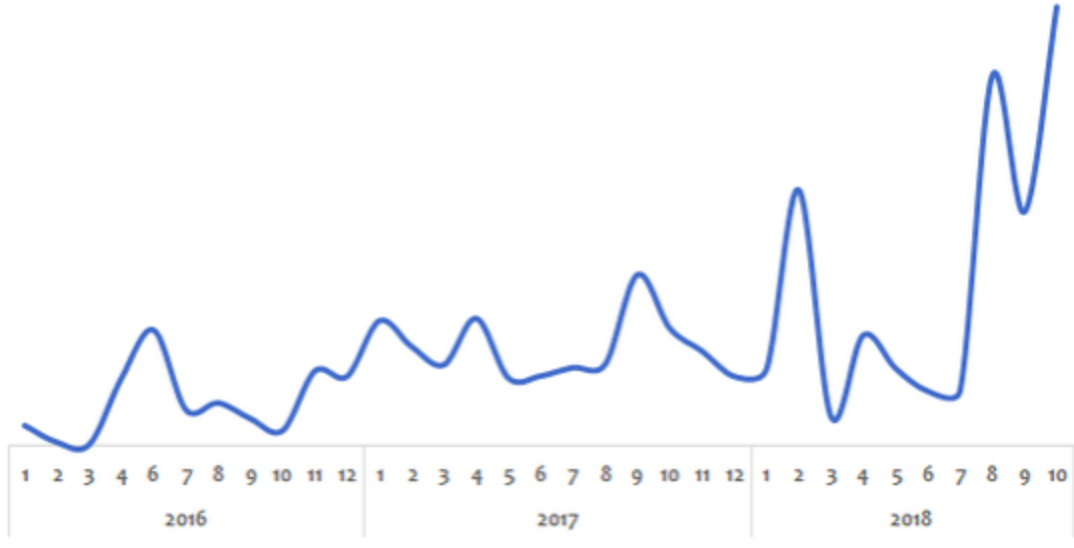
- Challenge:
  - Peak usage during holidays and weekends
  - Less load during rest of the time
- Solution (before the Cloud):
  - **Procure** (Buy) infrastructure **for peak load**
    - What would the infrastructure be doing during periods of low loads?

Before the Cloud - Example 2 - Startup



- Challenge:
  - It suddenly becomes popular.
  - How to handle the **sudden increase** in load?
- Solution (before the Cloud):
  - **Procure** (Buy) infrastructure assuming they would be successful
    - What if they are not successful?

Before the Cloud - Challenges



- High cost of procuring infrastructure
- Needs ahead of time planning (**Can you guess the future?**)
- Low infrastructure utilization (**PEAK LOAD** provisioning)
- Dedicated infrastructure maintenance team (**Can a startup afford it?**)

Silver Lining in the Cloud

- How about **provisioning (renting) resources** when you want them and releasing them back when you do not need them?
  - **On-demand resource provisioning**
  - Also called **Elasticity**

Cloud - Advantages

- Trade **"capital expense"** for **"variable expense"**
- Benefit from massive **economies of scale**
- Stop **guessing** capacity
- Stop spending money running and maintaining data centers
- **"Go global"** in minutes

Microsoft Azure

- One of the leading cloud service providers
- Provides 200+ services
- Reliable, secure and cost-effective
- The entire course is all about Azure. You will learn it as we go further.

Best path to learn Azure!

Advisor

Machine Learning

Cosmos DB

Azure DevOps

- Cloud applications make use of multiple Azure services.
- There is **no single path** to learn these services independently.
- HOWEVER, we've worked out a simple path!

Setting up Azure Account

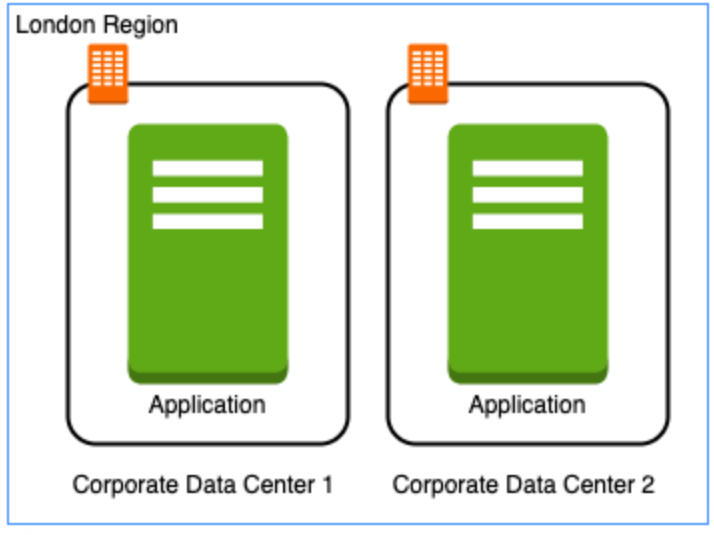
- Create Azure Account

Regions and Zones

Regions and Zones

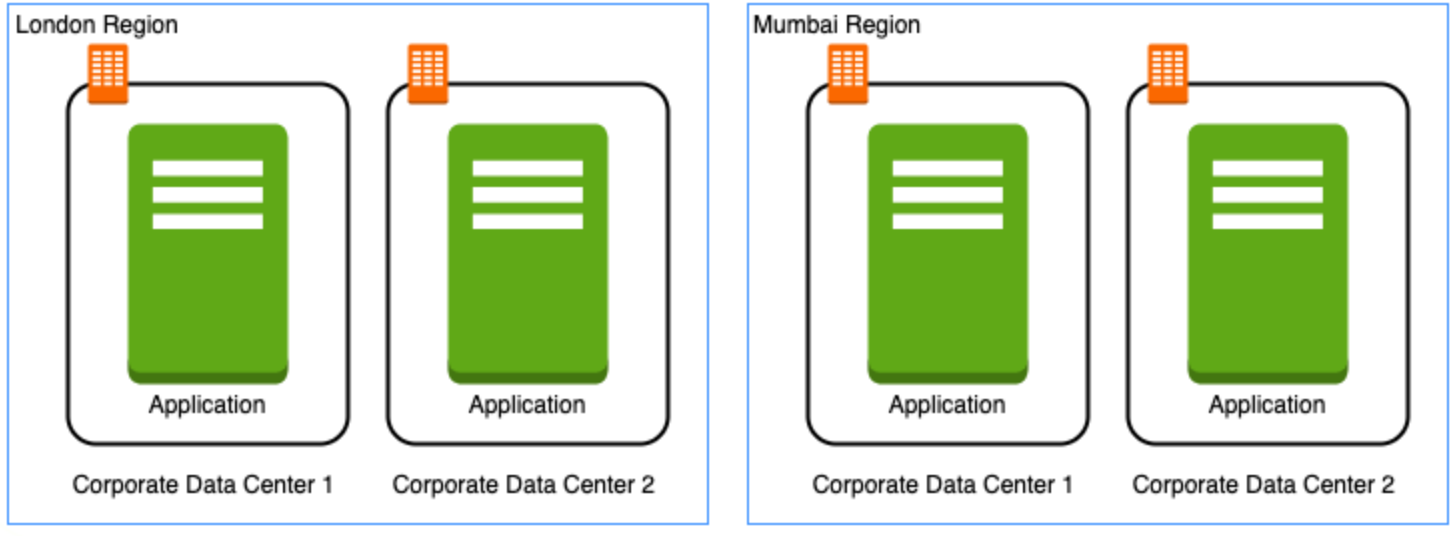
- Imagine that your application is deployed in a data center in London
- What would be the challenges?
  - Challenge 1 : Slow access for users from other parts of the world (**high latency**)
  - Challenge 2 : What if the data center crashes?
    - Your application goes down (**low availability**)

Multiple data centers



- Let's **add in one more data center** in London
- What would be the challenges?
  - Challenge 1 : Slow access for users from other parts of the world
  - Challenge 2 (**SOLVED**) : What if one data center crashes?
    - Your application is **still available** from the other data center
  - Challenge 3 : What if **entire region** of London is unavailable?
    - Your application goes down

Multiple regions



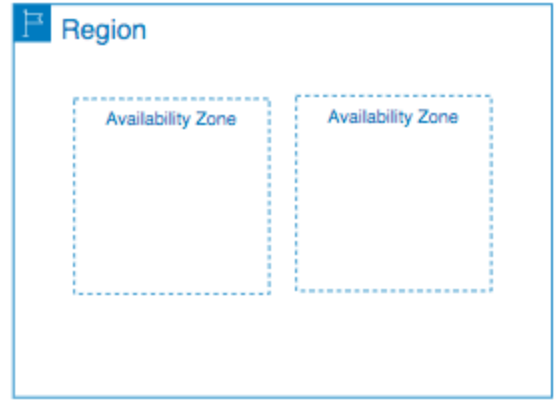
- Let's add a new region : Mumbai
- What would be the challenges?
  - Challenge 1 (**PARTLY SOLVED**) : Slow access for users from other parts of the world
    - You can solve this by adding deployments for your applications in other regions
  - Challenge 2 (SOLVED) : What if one data center crashes?
    - Your application is still live from the other data centers
  - Challenge 3 (**SOLVED**) : What if entire region of London is unavailable?
    - Your application is served from Mumbai

Regions

- Imagine setting up data centers in different regions around the world
  - **Would that be easy?**
- (Solution) Azure provides **60+ regions** around the world
  - Expanding every year
- **Region** : Specific geographical location to host your resources
- **Advantages**:
  - High Availability
  - Low Latency
  - Global Footprint
  - Adhere to government **regulations**

Availability Zones

- How to achieve high availability in the same region (or geographic location)?
  - Enter **Availability Zones**
    - Multiple AZs (3) in a region
    - **One or more discrete data centers**
    - Each AZ has **independent** & **redundant** power, networking & connectivity
    - AZs in a region are connected through **low-latency** links
- (Advantage) **Increased availability and fault tolerance** within same region
  - Survive the failure of a complete data center
- (Remember) NOT all Azure regions have Availability Zones



Regions and Availability Zones examples

New Regions and AZs are constantly added

Region	Availability Zones
(US) East US	3
(Europe) West Europe	3
(Asia Pacific) Southeast Asia	3
(South America) Brazil South	3
(US) West Central US	0

Azure Virtual Machines

Azure Virtual Machines

- In corporate data centers, applications are deployed to physical servers
- Where do you deploy applications in the cloud?
  - Rent virtual servers
  - **Virtual Machines** - Virtual servers in Azure
  - **Azure Virtual Machines** - Provision & Manage Virtual Machines

VM

Azure Virtual Machines - Features

- Create and manage lifecycle of Virtual Machine (VM) instances
- **Load balancing** and **auto scaling** for multiple VM instances
- **Attach storage** to your VM instances
- Manage **network connectivity and configuration** for your VM instances
- **Our Goal**:
  - Setup VM instances as HTTP (Web) Server
  - Distribute load with Load Balancers

VM

VM Scale Set

Azure Virtual Machines Hands-on

- Let's create a few VM instances and play with them
- Let's SSH into VM instances and install web server!

VM

Azure Virtual Machines - Key Concepts

Feature	Explanation
Image	Choose Operating System and Software
VM Family	Choose the right family of hardware (General purpose or Compute/Storage/Memory optimized or GPU or HPC)
VM Size (B1s, B2s, ...)	Choose the right quantity of hardware (2 vCPUs, 4GB of memory)
Disks	Attach Virtual Disks to VMs (Block Storage)

Useful Commands

```
#!/bin/sh
sudo su
apt-get -y update
apt-get -y install nginx
echo "Getting started with Azure Virtual Machines" > /var/www/html/index.html
echo "Welcome to in28minutes $(whoami)" > /var/www/html/index.html
echo "Welcome to in28minutes $(hostname)" > /var/www/html/index.html
```

- Commands:
  - `sudo su` - execute commands as a root user
  - `apt-get -y update` - Update package index - pull the latest changes from the repositories
  - `apt-get -y install nginx` - Install and start nginx web server
  - `echo "Hello World" > /var/www/html/index.html` - Write to index.html
  - `$(hostname)` - Get host name
  - `$(hostname -I)` - Get host internal IP address

Availability

- Are the applications available **when the users need them**?
  - **Percentage of time** an application provides the operations expected of it
- **Example**: 99.99% availability. Also called four 9's availability

Availability Table

Availability	Downtime (in a month)	Comment
99.95%	22 minutes	
99.99% (four 9's)	4 and 1/2 minutes	Most online apps aim for 99.99% (four 9's)
99.999% (five 9's)	26 seconds	Achieving 5 9's availability is tough

Increasing Availability for Azure VMs

- Single Instance VM:
  - Premium SSD or Ultra Disk: 99.9%
  - Standard SSD Managed Disks: 99.5%
  - Standard HDD Managed Disks: 95%
- Two or more instances in same Availability Set: 99.95%
  - Availability set is a logical grouping of VMs
    - **Fault domains**: Group of VMs sharing a common power source and network switch
    - **Update domains**: Group of VMs that are rebooted (updated) at the same time
- Two or more instances in two or more Availability Zones in the same Azure region: 99.99%
- **Summary**: Create multiple instances in multiple AZs if you want high availability

VM

Virtual Machine Scale Sets

- How do you simplify creation and management of multiple VMs?
  - Enter **Virtual machine scale sets**
- Allow you to create and manage a group of Azure VMs
  - Provides high availability to your applications
- (Optional) Add a load balancer
- (Optional) Distribute VM instances across Multiple AZs (where available)
- Supports Manual Scaling and Auto Scaling
- Supports up to 1,000 VM instances
- **DEMO TIME**

VM Scale Set

Azure Virtual Machines - More Features

Feature	Explanation
Static IP Address	Assign a fixed IP address to your VM Public IP addresses are charged per IP per hour
Azure Monitoring	Monitoring for your Azure VMs
Dedicated Hosts	Physical servers dedicated to one customer
Create cheaper, temporary instances for non critical workloads	Azure Spot instances
Reserve compute instances ahead of time	Reserved VM Instances (1 or 3 years)

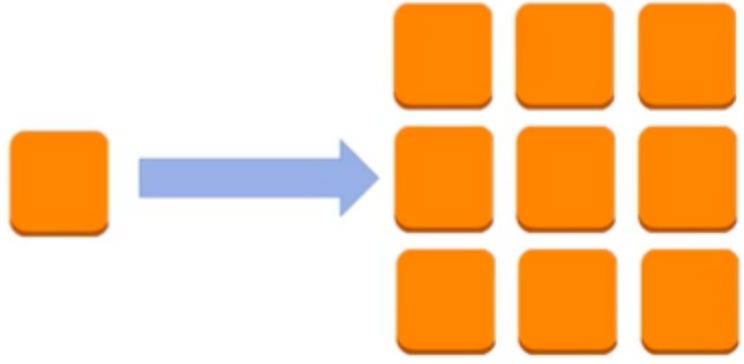
Designing Good Solutions with VMs

Terminology	Description	Azure VMs
Availability	Are the applications available **when your users need them**?	Availability Sets and Scale Sets
Scalability	Can we handle a **growth in users, traffic, or data size** without any drop in performance?	VM Size, Scale Sets and Load Balancers
Resilience	Ability of system to provide acceptable behavior even when one or more parts of the system fail	Scale Sets and Load Balancers
Geo-distribution	Distribute applications across regions and zones	Scale Sets and Load Balancers
Disaster Recovery	How to keep your systems running in face of disasters?	Site Recovery
Managing Costs	You want to keep costs low	Auto Scaling (Elasticity), Reservations, Spot Instances
Security	Secure your VMs	Dedicated Hosts, (More to come...)

Vertical Scaling

- Deploying application/database to **bigger instance**:
  - A larger hard drive
  - A faster CPU
  - More RAM, CPU, I/O, or networking capabilities
- In Azure: We can increase VM size
- There are limits to vertical scaling

Horizontal Scaling



- Deploying multiple instances of application/database
- (Typically but not always) Horizontal Scaling is preferred to Vertical Scaling:
  - Vertical scaling has limits
  - Vertical scaling can be expensive
  - Horizontal scaling increases availability
- (BUT) Horizontal Scaling needs additional infrastructure:
  - Scaling Sets, Load Balancers etc.

Azure Virtual Machines - Scenarios

Scenario	Solution
How can you automatically scale up and scale down VMs?	VM Scale Sets
How can you protect VMs from datacenter failures?	Deploy them to multiple AZs (Scale Sets)
How much availability do you get by deploying two or more VM instances in two or more AZs in same region?	99.99%
How can you perform disaster recovery for your VMs?	Site Recovery
How can you reduce costs for your VMs?	AutoScaling(Elasticity), Reserved & Spot Instances, Right Region - Cost varies from region to region
Will you be billed if you stop a VM?	Yes. For Storage.
Will two VMs of same size always cost the same?	No. Price changes with time. Price also is different in different regions.
How can you know who performed a specific action on a VM?	Activity Logs (kept for 90 days)

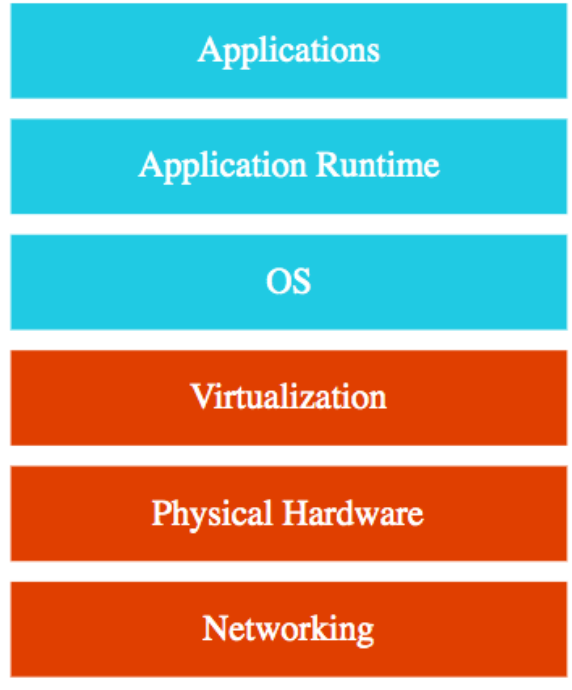
Managed Services

Managed Services

- Do you want to continue **running applications in the cloud**, the **same way you run them** in your **data center**?
- OR **are there OTHER approaches**?
- You should **understand some terminology** used with cloud services:
  - **IaaS** (Infrastructure as a Service)
  - **PaaS** (Platform as a Service)
  - **SaaS** (Software as a Service)
  - **Serverless**
- Let's get on a quick **journey** to understand these!

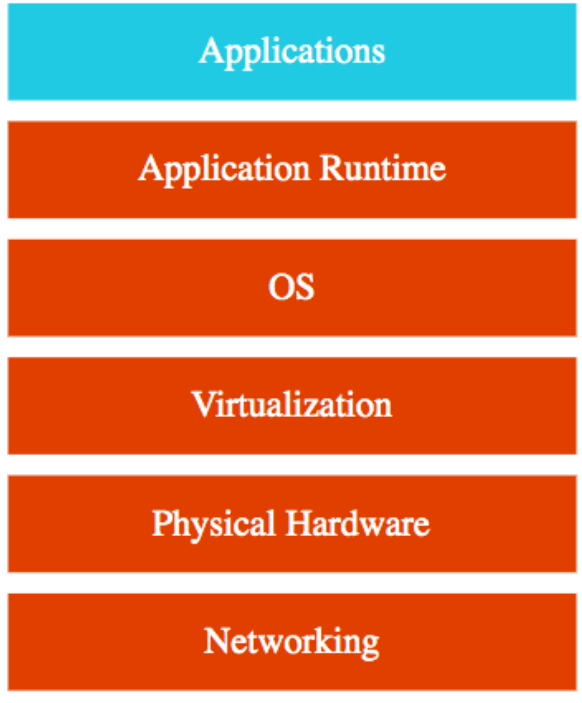
IAAS (Infrastructure as a Service)

- Use **only infrastructure** from cloud provider
- **Example**: Using VM to deploy your applications or databases
- You are responsible for:
  - Application Code and Runtime
  - Configuring load balancing
  - Auto scaling
  - OS upgrades and patches
  - Availability
  - etc.. (and a lot of things!)



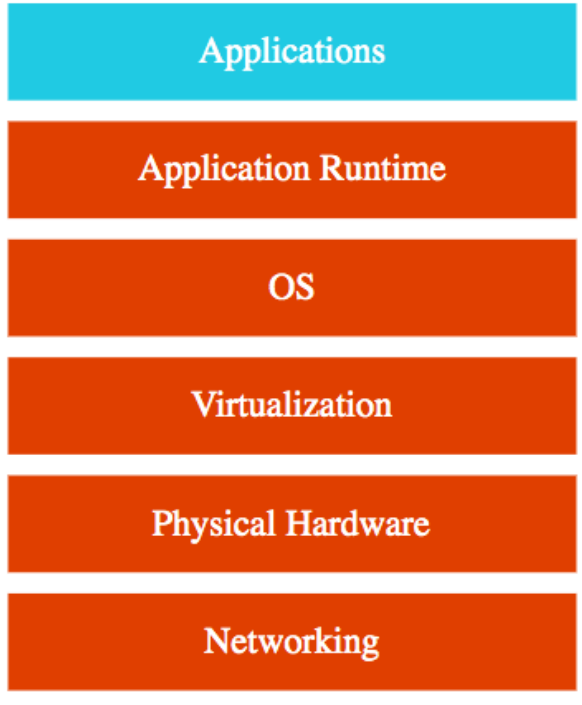
PAAS (Platform as a Service)

- Use a platform provided by cloud
- **Cloud provider** is responsible for:
  - OS (incl. upgrades and patches)
  - Application Runtime
  - Auto scaling, Availability & Load balancing etc..
- **You** are responsible for:
  - Configuration (of Application and Services)
  - Application code (if needed)
- Examples:
  - Azure App Service
  - Databases - Relational & NoSQL (Amazon RDS, Google Cloud SQL, Azure SQL Database etc)
  - Queues, AI, ML, Operations etc!

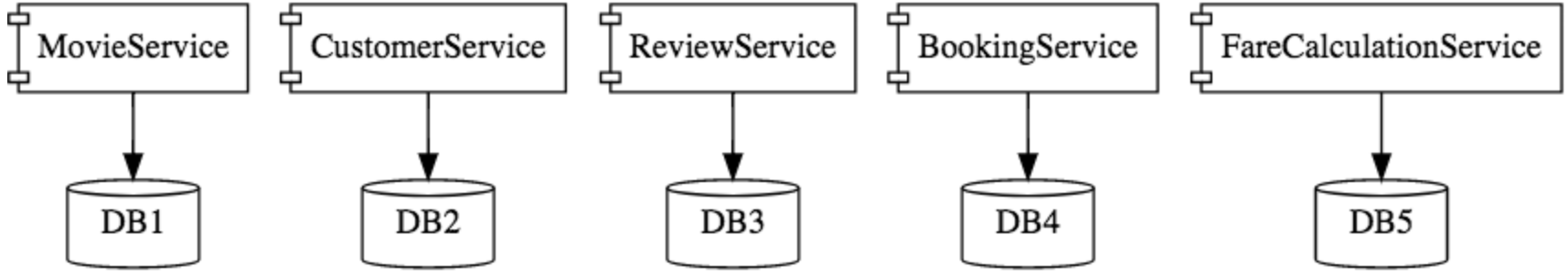


Azure App Service

- Fully managed platform for building, deploying and scaling your web apps
  - Also supports REST APIs, and mobile back ends
- Natively supports .NET, .NET Core, Node.js, Java, Python and PHP
- Choose App Service plan: defines a set of compute resources for a web app
- Features:
  - Automated Deployment and management
  - Auto Scaling
  - Built in Load Balancing



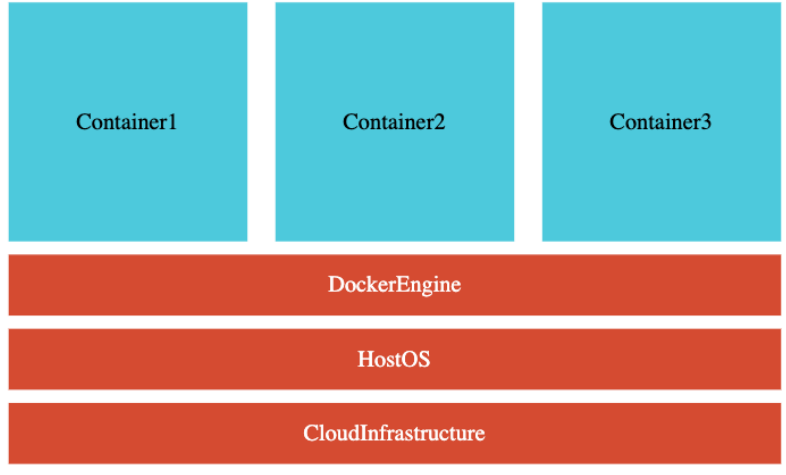
Microservices



- Enterprises are heading towards microservices architectures
  - Build small focused microservices
  - **Flexibility to innovate** and build applications in different programming languages (Go, Java, Python, JavaScript, etc)
- BUT **deployments become complex**!
- How can we have **one way of deploying** Go, Java, Python or JavaScript .. microservices?
  - Enter **containers**!

Containers - Docker

- Create **Docker images** for each microservice
- Docker image **has all needs of a microservice**:
  - Application Runtime (JDK or Python or NodeJS)
  - Application code and Dependencies
  - VMs virtualize Hardware while containers virtualize OS
  - Runs **the same way** on any infrastructure:
    - Your local machine
    - Corporate data center
    - Cloud
- Advantages
  - Docker containers are **light weight**
    - Compared to Virtual Machines as they do not have a Guest OS
  - Docker provides **isolation** for containers
  - Docker is **cloud neutral**

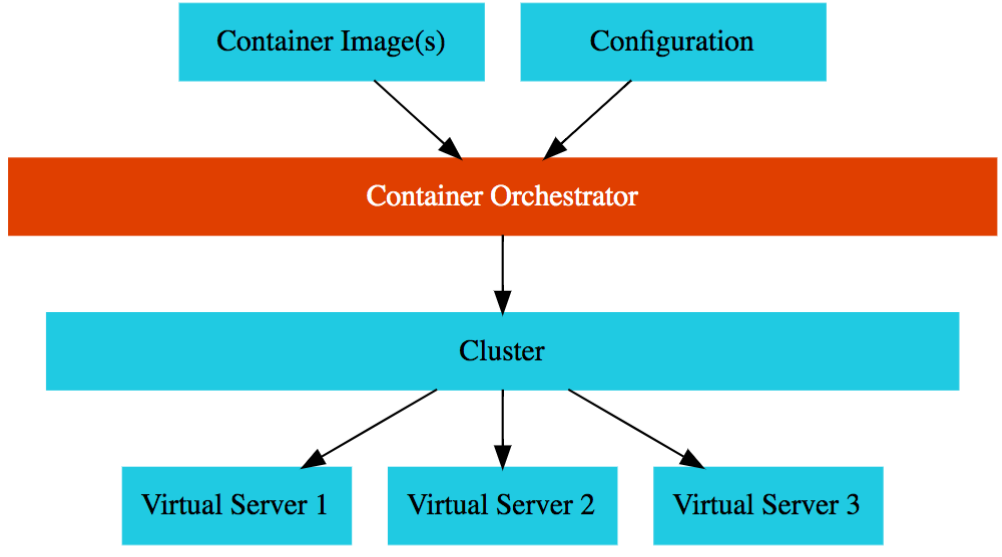


Azure Container Instances

- Manage and run simple container based applications
- **You DO NOT** need to provision and manage VMs
- Start containers in seconds
- Azure App Service also supports deploying simple containers

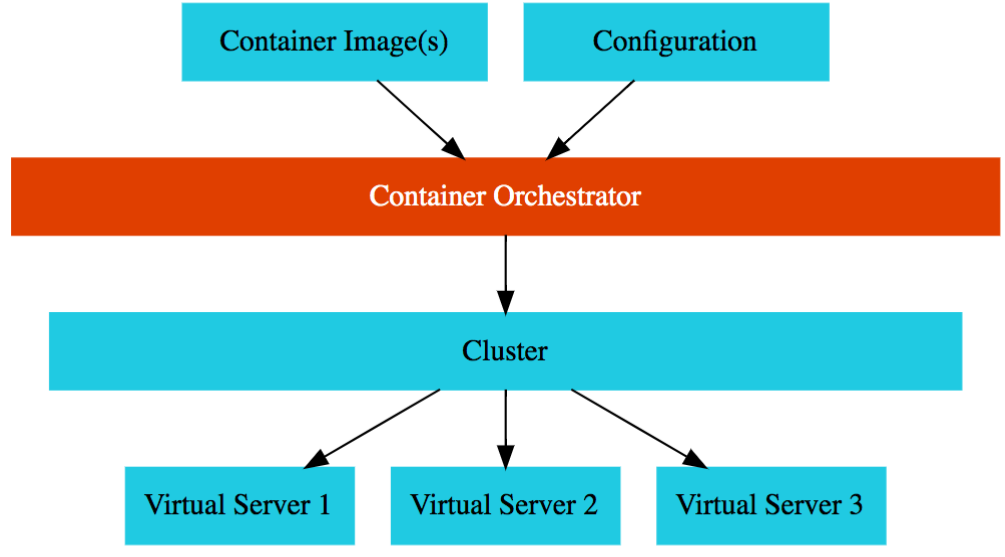
Container Orchestration

- **Requirement** : I want 10 instances of Microservice A container, 15 instances of Microservice B container and ....
- Typical Features:
  - **Auto Scaling** - Scale containers based on demand
  - **Service Discovery** - Help microservices find one another
  - **Load Balancer** - Distribute load among multiple instances of a microservice
  - **Self Healing** - Do health checks and replace failing instances
  - **Zero Downtime Deployments** - Release new versions without downtime



Container Orchestration - AKS and Service Fabric

- Using a Container Orchestrator:
  - 1: Create a Cluster
  - 2: Deploy & Orchestrate Microservices
- Azure Services:
  - Azure Kubernetes Service: Managed Kubernetes Service
  - Azure Service Fabric: Microsoft's container orchestrator



Serverless

- What do we think about when we develop an application?
  - Where to deploy? What kind of server? What OS?
  - How do we take care of scaling and availability of the application?
- **What if you don't need to worry about servers and focus on your code?**
  - Enter **Serverless**
    - Remember: **Serverless does NOT mean "No Servers"**
- **Serverless for me:**
  - You **don't worry** about infrastructure (ZERO visibility into infrastructure)
    - Flexible scaling and automated high availability
  - Most Important: **Pay for use**
    - Ideally ZERO REQUESTS => ZERO COST
- **You focus on code** and the cloud managed service takes care of all that is needed to scale your code to serve millions of requests!
  - And you pay for requests and NOT servers!

Azure Functions

- You don't worry about servers or scaling or availability
- You only worry about your code
- You pay for what you use
  - Number of requests
  - Duration of requests
  - Memory consumed
- Supports C#, Python, JavaScript, Typescript and Java

SaaS (Software as a Service)

- **Centrally hosted software** (mostly on the cloud)
  - Offered on a subscription basis (pay-as-you-go)
  - Examples:
    - Email, calendaring & office tools (such as Outlook 365, Microsoft Office 365, Gmail, Google Docs)
    - Customer relationship management (CRM), enterprise resource planning (ERP) and document management tools
- **Cloud provider** is responsible for:
  - OS (incl. upgrades and patches)
  - Application Runtime
  - Auto scaling, Availability & Load balancing etc..
  - Application code and/or
  - Application Configuration (How much memory? How many instances? ..)
- **Customer** is responsible for:
  - Configuring the software!

Shared responsibility model

Azure Cloud Service Categories - Scenarios

Scenario	Solution
IaaS or PaaS or SaaS: Deploy Custom Application in Virtual Machines	IaaS
IaaS or PaaS or SaaS: Using Gmail	SaaS
IaaS or PaaS or SaaS: Using Azure App Service to deploy your app	PaaS
True or False: Customer is responsible for OS updates when using PaaS	False
True or False: Customer is responsible for Availability when using PaaS	False
True or False: In PaaS, customer has access to VM instances	False
True or False: In PaaS, customer can customize OS and install custom software	False
True or False: In PaaS, customer can configure auto scaling needs	True
True or False: In PaaS, customer can configure hardware needs (memory, cpu etc)	True
True or False: PaaS services only offer Compute services	False

Review - Azure Services for Compute

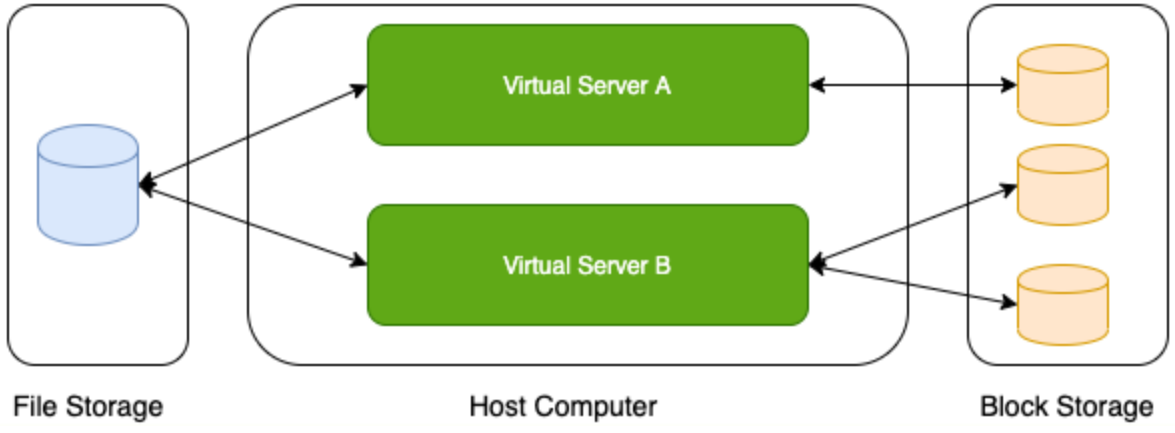
Azure Service Name	Description
Azure VMs	Windows or Linux VMs (IaaS) Use VMs when you need control over OS OR you want to run custom software You handle Availability, Scalability, Load Balancing, Software/OS Updates ...
Azure App Service	PaaS. Deploy web apps, mobile back ends and RESTful APIs quickly. Built-in Auto Scaling, Load Balancing
Azure Container Instances	PaaS (CaaS). Run isolated containers, without orchestration. You DO NOT need to provision and manage VMs. Start containers in seconds.
Azure Kubernetes Service	PaaS (CaaS). Managed Kubernetes Service. Provides container orchestration.
Azure Service Fabric	PaaS (CaaS). Microsoft's container orchestrator. Package, deploy, and manage scalable and reliable microservices Run anywhere - on premises and in the cloud
Azure Functions	Serverless (FaaS) compute for event-driven apps

Azure Compute Services - Scenarios

Scenario	Solution
You want to run function in response to events	Azure Functions
You want to deploy a Python application using a Managed Service	Azure App Service
You want to quickly deploy a container	Azure Container Instances
You want to setup a complex microservices architecture in Azure	AKS or Service Fabric
Your application needs customized OS and custom Software installed	Azure VMs

Storage

Storage Types - Block, File, Object, ……



- What is the type of storage of your hard disk?
  - **Block Storage**
- You've created a file share to share a set of files with your colleagues in a enterprise. What type of storage are you using?
  - **File Storage**
- You want to be able to upload/download objects using a REST API without mounting them onto your VM. What type of storage are you using?
  - **Object Storage**

Azure Storage

- Managed Cloud Storage Solution
  - Highly available, durable and massively scalable (upto few PetaBytes)
- Core Storage Services:
  - **Azure Disks**: Block storage (hard disks) for Azure VMs
  - **Azure Files**: File shares for cloud and on-premises
  - **Azure Blobs**: Object store for text and binary data
  - **Azure Queues**: Decouple applications using messaging
  - **Azure Tables**: NoSQL store (Very Basic)
    - Prefer Azure Cosmos DB for NoSQL
- (PRE-REQUISITE) Storage Account is needed for Azure Files, Azure Blobs, Azure Queues and Azure Tables

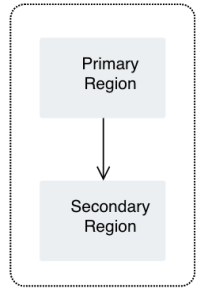
Azure Storage

Azure Storage - Data Redundancy

Option	Redundancy	Discussion
Locally redundant storage (LRS)	Three synchronous copies in same data center	Least expensive and least availability
Zone-redundant storage (ZRS)	Three synchronous copies in three AZs in the primary region	
Geo-redundant storage (GRS)	LRS + Asynchronous copy to secondary region (three more copies using LRS)	
Geo-zone-redundant storage (GZRS)	ZRS + Asynchronous copy to secondary region (three more copies using LRS)	Most expensive and highest availability

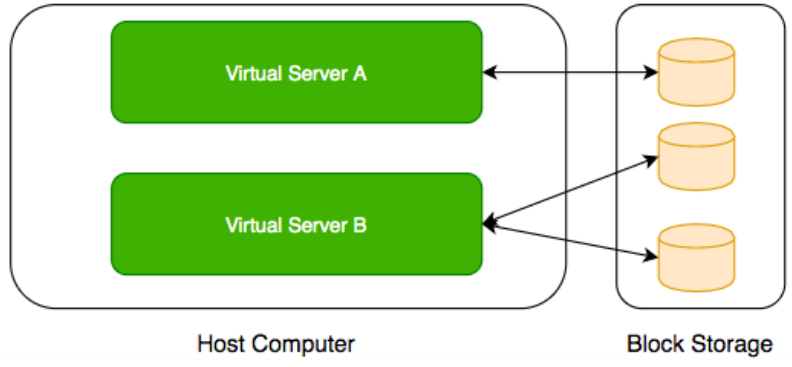
Region pairs

- Data copies across regions => high availability + high durability
- Azure makes it easy to distribute data across regions (while retaining data in same geography) through **Region Pairs**
  - **Examples**: Central India & South India, East US & West US, North Europe (Ireland) & West Europe (Netherlands), ..
  - **Azure Storage Example**: If you use Geo-redundant storage (GRS) and choose region as East US
    - 3 copies stored in East US and 3 copies in the corresponding paired region - West US
    - Access data from primary region (East US)
    - Option to failover to secondary region (West US) if primary region is NOT available
- Region pairs have **very fast data connection**
- Azure tries to ensure that both regions (in a region pair) **do NOT have problems at the same time**
  - For Example: Software updates are done one region at a time



Block Storage

- Use case: Hard-disks attached to your computers
- Typically, ONE Block Storage device can be connected to ONE virtual server
- HOWEVER, you can connect multiple different block storage devices to one virtual server

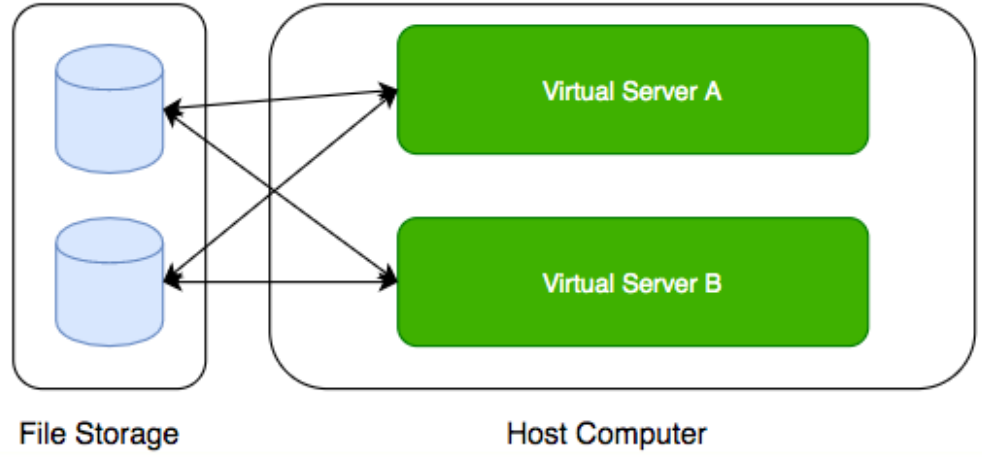


Azure Disks Storage

- **Disk storage**: Disks for Azure VMs
  - **Types**:
    - **Standard HDD**: Recommended for Backup, non-critical, infrequent access
    - **Standard SSD**: Recommended for Web servers, lightly used enterprise applications and dev/test environments
    - **Premium SSD disks**: Recommended for production and performance sensitive workloads
    - **Ultra disks (SSD)**: Recommended for IO-intensive workloads such as SAP HANA, top tier databases (for example, SQL, Oracle), and other transaction-heavy workloads
  - Premium and Ultra provide very high availability
- **Managed vs Unmanaged Disks**:
  - Managed Disks are easy to use:
    - Azure handles storage
    - High fault tolerance and availability
  - Unmanaged Disks are old and tricky (Avoid them if you can)
    - You need to manage storage and storage account
    - Disks stored in Containers (NOT Docker containers. Completely unrelated.)

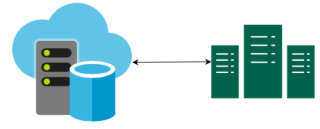
Azure Files

- Media workflows need huge shared storage for things like video editing
- Enterprise users need a quick way to share files in a secure & organized way
- **Azure Files**:
  - Managed File Shares
  - Connect from multiple devices concurrently:
    - From cloud or on-premises
    - From different OS: Windows, Linux, and macOS
  - Supports Server Message Block (SMB) and Network File System (NFS) protocols
  - Usecase: Shared files between multiple VMs (example: configuration files)



Azure File Sync

- **Windows file server**: Create file shares on-premises
- **Azure Files**: Create file shares on Azure
- Storing files in Azure Files is cheaper & easier to manage BUT Windows file server provides flexible connectivity options to on-premise apps and users
  - How about **having same connectivity to file shares for on-premises apps and resources** while storing them in Azure Files?
- **Azure File Sync**: File shares created in Azure Files. AND Retain flexibility and compatibility of Windows file server.
  - **Option**: Keep cache of frequently accessed files or have a full local copy
  - **Supports multiple protocols**: SMB, NFS, and FTPS
  - **Advantages**: Cheaper, easier to manage and can be used as cloud-side backup (Business continuity and disaster recovery)



Azure Blob Storage

- **Azure Blob Storage**: Object storage in Azure
- **Structure**: Storage Account > Container(s) > Blob(s)
- Store massive volumes of unstructured data
  - **Store all file types** - text, binary, backup & archives:
    - Media files and archives, Application packages and logs
    - Backups of your databases or storage devices
- **Three Types of Blobs**
  - Block Blobs: Store text or binary files (videos, archives etc)
  - Append Blobs: Store log files (Ideal for append operations)
  - Page Blobs: Foundation for Azure IaaS Disks (512-byte pages up to 8 TB)
- **Azure Data Lake Storage Gen2**: Azure Blob Storage Enhanced
  - Designed for enterprise big data analytics (exabytes, hierarchical)
  - Low-cost, tiered storage, with high availability/disaster recovery

Azure Storage

Azure Blob Storage - Access Tiers

- **Different kinds of data** can be stored in Blob Storage
  - Media files, website static content
  - Backups of your databases or storage devices
  - Long term archives
- Huge variations in **access patterns**
- Can I pay a cheaper price for objects I access less frequently?
  - **Access tiers**
    - **Hot**: Store frequently accessed data
    - **Cool**: Infrequently accessed data stored for min. 30 days
    - **Archive**: Rarely accessed data stored for min. 180 days
      - Lowest storage cost BUT Highest access cost
      - Access latency: In hours
      - To access: **Rehydrate** (Change access tier to hot or cool) OR
        - Copy to another blob with access tier hot or cool
    - You can **change access tiers** of an object **at any point in time**

Azure Storage

Azure Queues and Tables

- **Azure Queues**: Decouple applications using messaging
- **Azure Tables**: NoSQL store (Very Basic)
  - Prefer Azure Cosmos DB for NoSQL

Azure Storage Explorer

- **Azure Storage Explorer**: Manage Azure storage resources from desktop
  - **Free tool**: Supported on Windows, macOS, and Linux
    - **Integrates with**:
      - Azure Storage blobs, files, queues, and tables
      - Azure Data Lake Storage
      - Azure managed disks
  - **Features**: Upload, download files, manage permissions, ..
  - **Extensions available**: Data Factory extn - move data from AWS S3 to Azure Storage
  - Very similar to **Storage Explorer** and **Storage Browser** on Azure Portal
- (**Alternative**) **AzCopy**: Command-line utility
  - Copy files from local machine or other cloud storage to Azure Storage
  - (REMEMBER) Azure Storage Explorer uses AzCopy in the background
    - Use Azure Storage Explorer if you prefer a GUI
    - Use AzCopy if you like command line or you want to automate

Database Fundamentals

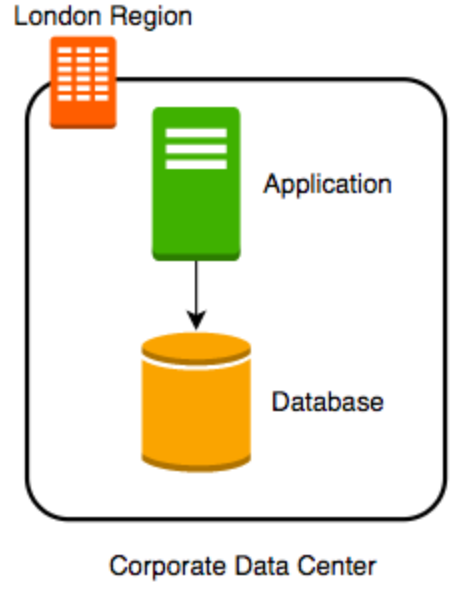
Databases Primer

- Databases provide **organized** and **persistent** storage for your data
- To **choose between different database types**, we would need to understand:
  - Availability
  - Durability
  - RTO
  - RPO
  - Consistency
  - Transactions etc
- Let's get started on a **simple journey** to understand these

Database

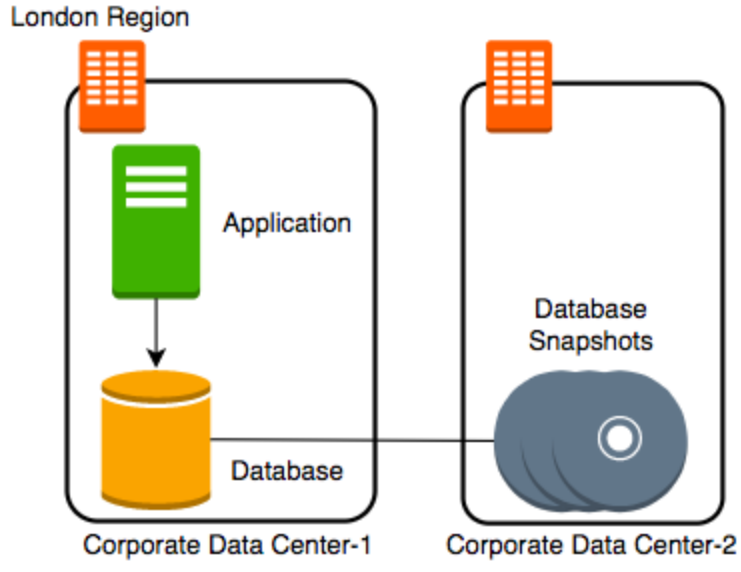
Database - Getting Started

- Imagine a database deployed **in a data center in London**
- Let's consider some challenges:
  - **Challenge 1**: Your database will go down if the data center crashes or the server storage fails
  - **Challenge 2**: You will lose data if the database crashes



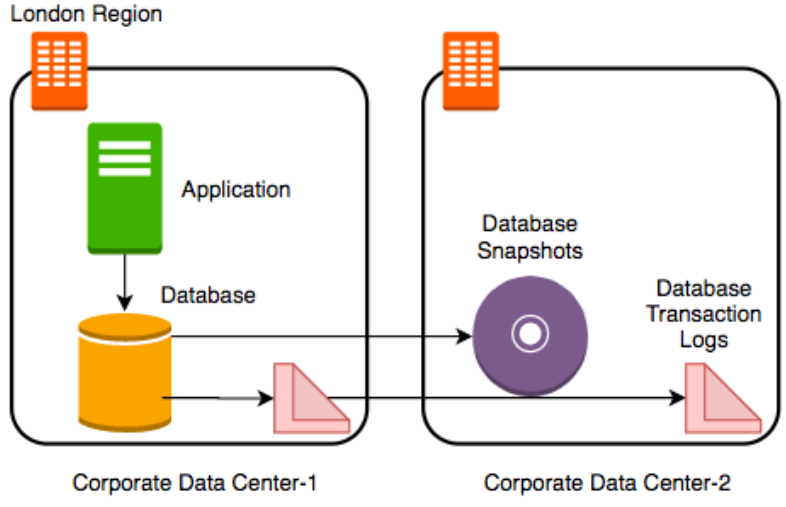
Database - Snapshots

- Let's automate taking copy of the database (**take a snapshot**) every hour to another data center in London
- Let's consider some challenges:
  - **Challenge 1**: Your database will go down if the data center crashes
  - **Challenge 2** (PARTIALLY SOLVED): You will lose data if the database crashes
    - You can setup database from latest snapshot. But depending on when failure occurs you can lose up to an hour of data
  - **Challenge 3**(NEW): Database will be slow when you take snapshots



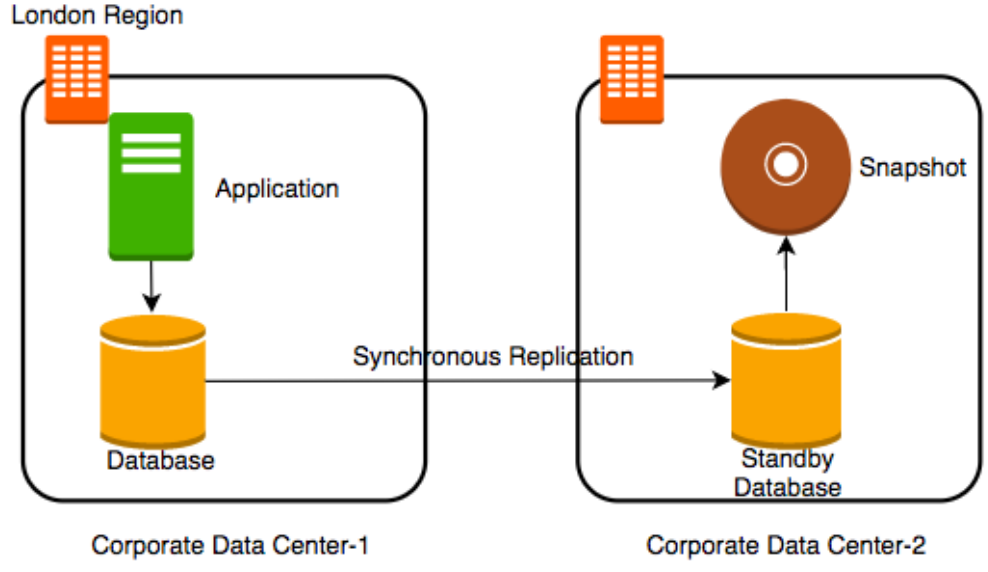
Database - Transaction Logs

- Let's add **transaction logs** to database and create a **process to copy it over** to the second data center
- Let's consider some challenges:
  - **Challenge 1**: Your database will go down if the data center crashes
  - **Challenge 2** (SOLVED): You will lose data if the database crashes
    - You can setup database from latest snapshot and apply transaction logs
  - **Challenge 3**: Database will be slow when you take snapshots



Database - Add a Standby

- Let's add a **standby database** in the second data center with replication
- Let's consider some challenges:
  - **Challenge 1** (SOLVED): Your database will go down if the data center crashes
    - You can switch to the standby database
  - **Challenge 2** (SOLVED): You will lose data if the database crashes
  - **Challenge 3** (SOLVED): Database will be slow when you take snapshots
    - Take snapshots from standby.
    - Applications connecting to master will get good performance always



Availability and Durability

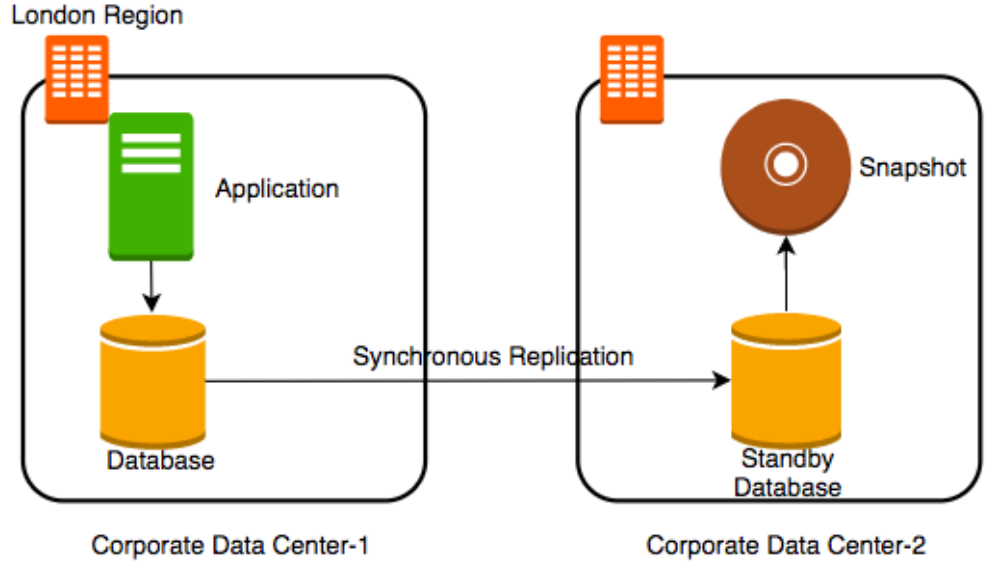
- **Availability**
  - Will I be able to access my data now and when I need it?
  - Percentage of time an application provides the operations expected of it
- **Durability**
  - Will my data be available after 10 or 100 or 1000 years?
- Examples of measuring availability and durability:
  - 4 9's - 99.99
  - 11 9's - 99.999999999
- Typically, an **availability of four 9's** is considered very good
- Typically, a **durability of eleven 9's** is considered very good

Availability

Availability	Downtime (in a month)	Comment
99.95%	22 minutes	
99.99% (4 9's)	4 and 1/2 minutes	Typically online apps aim for 99.99% (4 9's) availability
99.999% (5 9's)	26 seconds	Achieving 5 9's availability is tough

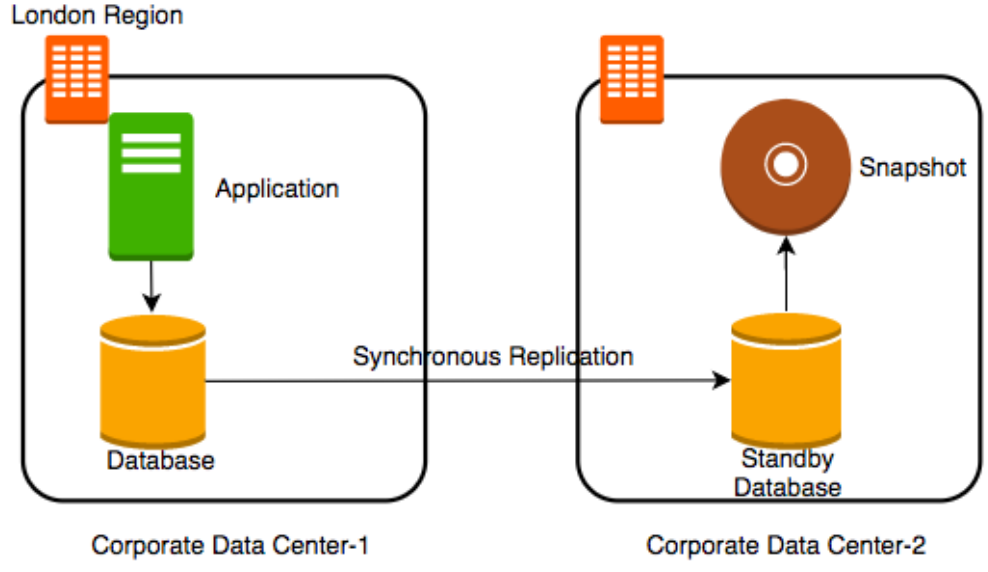
Durability

- What does a **durability of 11 9's** mean?
  - If you **store one million files for ten million years**, you would expect to **lose one file**
- Why should durability be high?
  - Because **we hate losing data**
  - Once we lose data, it is gone



Increasing Availability and Durability of Databases

- **Increasing Availability**:
  - Have multiple standbys available OR distribute the database
    - in multiple Zones
    - in multiple Regions
- **Increasing Durability**:
  - Multiple copies of data (standbys, snapshots, transaction logs and replicas)
    - in multiple Zones
    - in multiple Regions
- **Replicating data** comes with its own challenges!
  - We will talk about them a little later



Database Terminology : RTO and RPO

- Imagine a **financial transaction being lost**
- Imagine a **trade being lost**
- Imagine a **stock exchange going down** for an hour
- **Typically** businesses are fine with some downtime but they hate losing data
- Availability and Durability are technical measures
- How do we measure **how quickly we can recover from failure**?
  - **RPO (Recovery Point Objective)**: Maximum acceptable period of data loss
  - **RTO (Recovery Time Objective)**: Maximum acceptable downtime
- Achieving **minimum RTO and RPO is expensive**
- **Trade-off** based on the criticality of the data

Database

Question - RTO and RPO

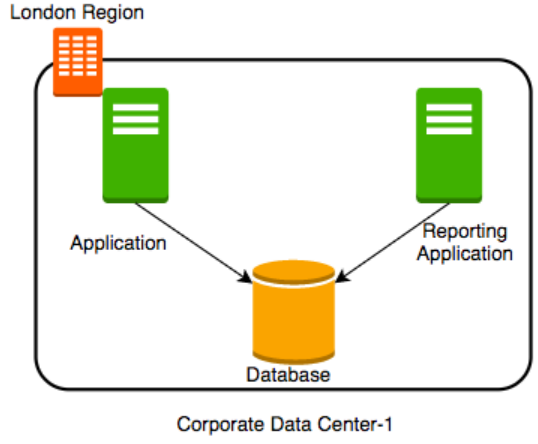
- You are running an application in VM instance storing its data on a persistent data storage. You are taking snapshots every 48 hours. If the VM instance crashes, you can manually bring it back up in 45 minutes from the snapshot. What is your RTO and RPO?
  - RTO - 45 minutes
  - RPO - 48 hours

Achieving RTO and RPO - Failover Examples

Scenario	Solution
Very small data loss (RPO - 1 minute) **Very small downtime (RTO - 5 minutes)**	**Hot standby** - Automatically synchronize data Have a standby ready to pick up load Use automatic failover from master to standby
Very small data loss (RPO - 1 minute) **BUT I can tolerate some downtimes (RTO - 15 minutes)**	**Warm standby** - Automatically synchronize data Have a standby with minimum infrastructure Scale it up when a failure happens
Data is critical (RPO - 1 minute) but I can tolerate downtime of a few hours (RTO - few hours)	Create regular data **snapshots and transaction logs** Create database from snapshots and transactions logs when a failure happens
Data can be lost without a problem (for example: cached data)	Failover to a completely new server

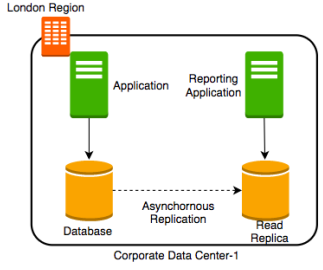
(New Scenario) Reporting and Analytics Applications

- New reporting and analytics applications are being launched using the same database
  - These applications will ONLY read data
- Within a few days you see that the database performance is impacted
- How can we fix the problem?
  - **Vertically scale the database** - increase CPU and memory
  - **Create a database cluster (Distribute the database)** - Typically database clusters are expensive to setup
  - **Create read replicas** - Run read only applications against read replicas



Consistency

- How do you ensure that data in multiple database instances (standbys and replicas) is updated simultaneously?
- **Strong consistency** - Synchronous replication to all replicas
  - Will be slow if you have multiple replicas or standbys
- **Eventual consistency** - Asynchronous replication. A little lag - few seconds - before the change is available in all replicas
  - In the intermediate period, different replicas might return different values
  - Used when scalability is more important than data integrity
  - Examples : Social Media Posts - Facebook status messages, Twitter tweets, Linked in posts etc
- **Read-after-Write consistency** - Inserts are immediately available
  - However, updates would have eventual consistency



Database Categories

- There are **several categories** of databases:
  - Relational (OLTP and OLAP), Document, Key Value, Graph, In Memory among others
- **Choosing type of database** for your use case is not easy. A few factors:
  - Do you want a **fixed schema**?
    - Do you want flexibility in defining and changing your schema? (schemaless)
  - What level of **transaction properties** do you need? (atomicity and consistency)
  - What kind of **latency** do you want? (seconds, milliseconds or microseconds)
  - **How many transactions** do you expect? (hundreds or thousands or millions of transactions per second)
  - **How much data** will be stored? (MBs or GBs or TBs or PBs)
  - and a lot more...

SQL Database

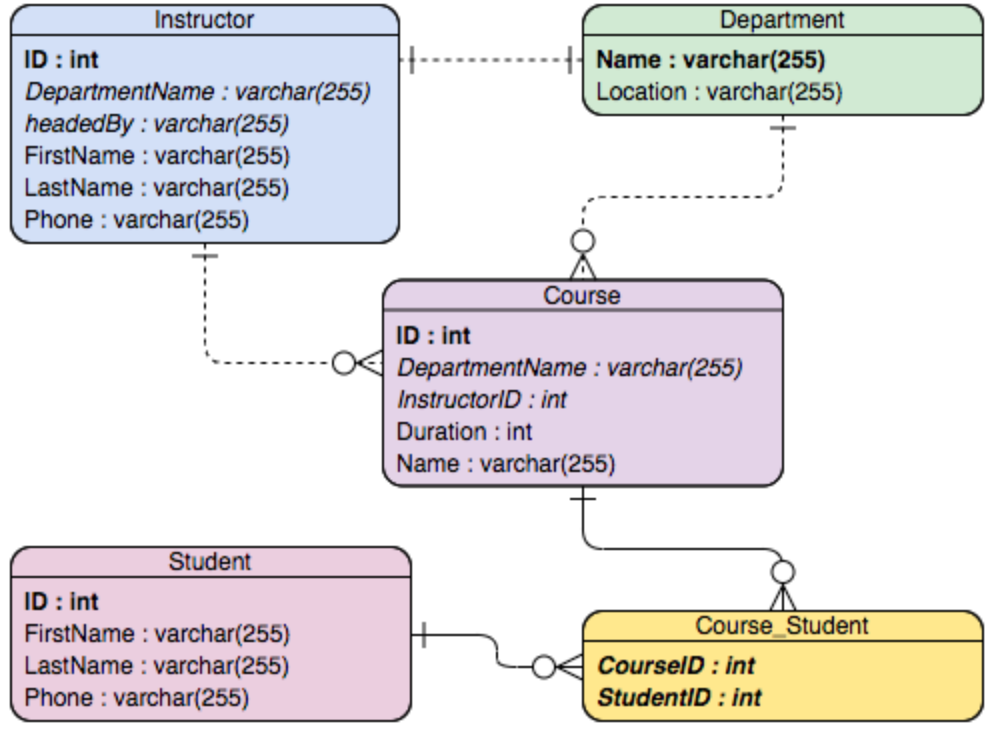

Cosmos DB


Azure Database PostgreSQL


Synapse Analytics

Relational Databases

- This was the **only option** until a decade back!
- Most **popular (or unpopular)** type of databases
- **Predefined schema** with tables and relationships
- Very **strong transactional** capabilities
- Used for
  - OLTP (Online Transaction Processing) use cases and
  - OLAP (Online Analytics Processing) use cases



Relational Database - OLTP (Online Transaction Processin

- Applications where **large number of users make large number of small transactions**
  - small data reads, updates and deletes
- **Use cases:**Most traditional applications - ERP, CRM, e-commerce, banking
- **Popular databases**:
  - MySQL, Oracle, SQL Server etc
- **Recommended Azure Managed Services**:
  - **Azure SQL Database**: Managed Microsoft SQL Server
  - **Azure Database for MySQL**: Managed MySQL
  - **Azure Database for PostgreSQL**: Managed PostgreSQL

SQL Database

Azure Database PostgreSQL

Azure SQL Database

- **Fully Managed Service** for Microsoft SQL Server
- 99.99% availability
- **Built-in** high availability, automatic updates and backups
- Flexible and responsive serverless compute
- Hyperscale (up to 100 TB) storage

SQL Database

Azure database for MySQL

- Fully managed, scalable **MySQL database**
- Supports 5.6, 5.7 and 8.0 community editions of MySQL
- 99.99% availability
  - Choose single zone or zone redundant high availability
- Automatic updates and backups
- Typically used as part of **LAMP (Linux, Apache, MySQL, PHP/Perl/Python)** stack

Azure Database MySQL

Azure Database for PostgreSQL

- Fully managed, intelligent and scalable **PostgreSQL**
- 99.99% availability
  - Choose single zone or zone redundant high availability
- Automatic updates and backups
- **Single Server and Hyperscale Options**
  - Hyperscale: Scale to hundreds of nodes and execute queries across multiple nodes

Azure Database PostgreSQL

Relational Database - OLAP (Online Analytics Processing)

- Applications allowing users to **analyze petabytes of data**
  - **Examples** : Reporting applications, Data ware houses, Business intelligence applications, Analytics systems
  - **Sample application** : Decide insurance premiums analyzing data from last hundred years
  - Data is consolidated from multiple (transactional) databases
- Recommended Azure Managed Service
  - **Azure Synapse Analytics**: **Petabyte-scale** distributed data ware house
    - Provides a unified experience for developing end-to-end analytics solutions - Data integration + Enterprise data warehousing + Big data analytics
    - Enables MPP (massively parallel processing)
    - Run complex queries across petabytes of data
    - Earlier called Azure SQL Data Warehouse

Synapse Analytics

Relational Databases - OLAP vs OLTP

- OLAP and OLTP use **similar data structures**
- BUT **very different approach in how data is stored**
- **OLTP databases** use row storage
  - Each table row is stored together
  - Efficient for processing small transactions
- **OLAP databases** use columnar storage
  - Each table column is stored together
  - **High compression** - store petabytes of data efficiently
  - **Distribute data** - one table in multiple cluster nodes
  - **Execute single query across multiple nodes** - Complex queries can be executed efficiently