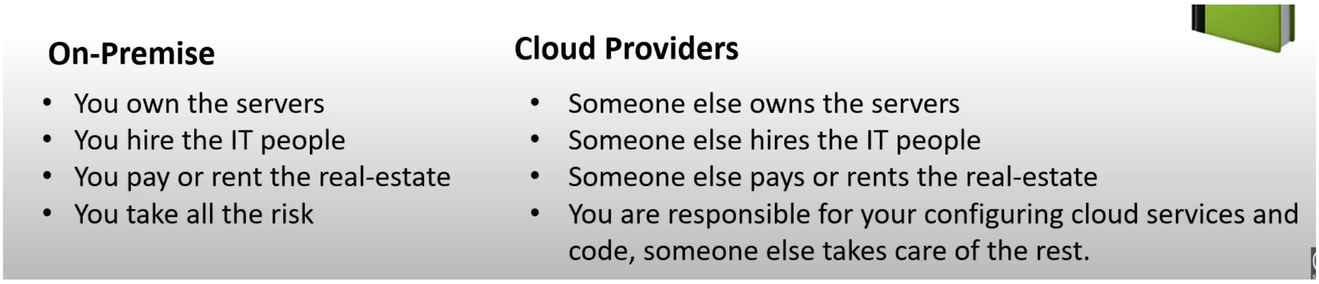
1. Cloud Concepts

Cloud computing (Absolute Basics)

The practice of using a network of remote servers hosted on the internet to :

1. Store
2. Manage
3. Process Data

instead of our own local systems.



On-Premise

- You own the servers
- You hire the IT people
- You pay or rent the real-estate
- You take all the risk

Cloud Providers

- Someone else owns the servers
- Someone else hires the IT people
- Someone else pays or rents the real-estate
- You are responsible for your configuring cloud services and code, someone else takes care of the rest.

Evolution of Cloud hosting

1. Dedicated Server

One physical machine, assigned to a single business.
Runs a *single web-app/website*

characteristics:
1. Very expensive
2. high maintenance
3. BUT ALSO HIGH SECURITY

2. VPS (Virtual Private Server)

One physical machine, dedicated to a single business (again)

> However, the key difference here is that the machine is *virtualised* into sub machines.

Each sub machine can run a web-app/ website, effectively meaning that multiple services can be ran on one single machine.

characteristics:
1. Better utilization of resources
2. Better isolation of the resources

3. Shared Hosting

Ex: Godaddy/ hoststaker

One physical machine, shared by *Hundreds* of businesses.

Relies on the tenants *under-utilising* their resources. Instead of a full fledged sub-machine being available to the tenants, it would instead be something of a folder-space that was given.

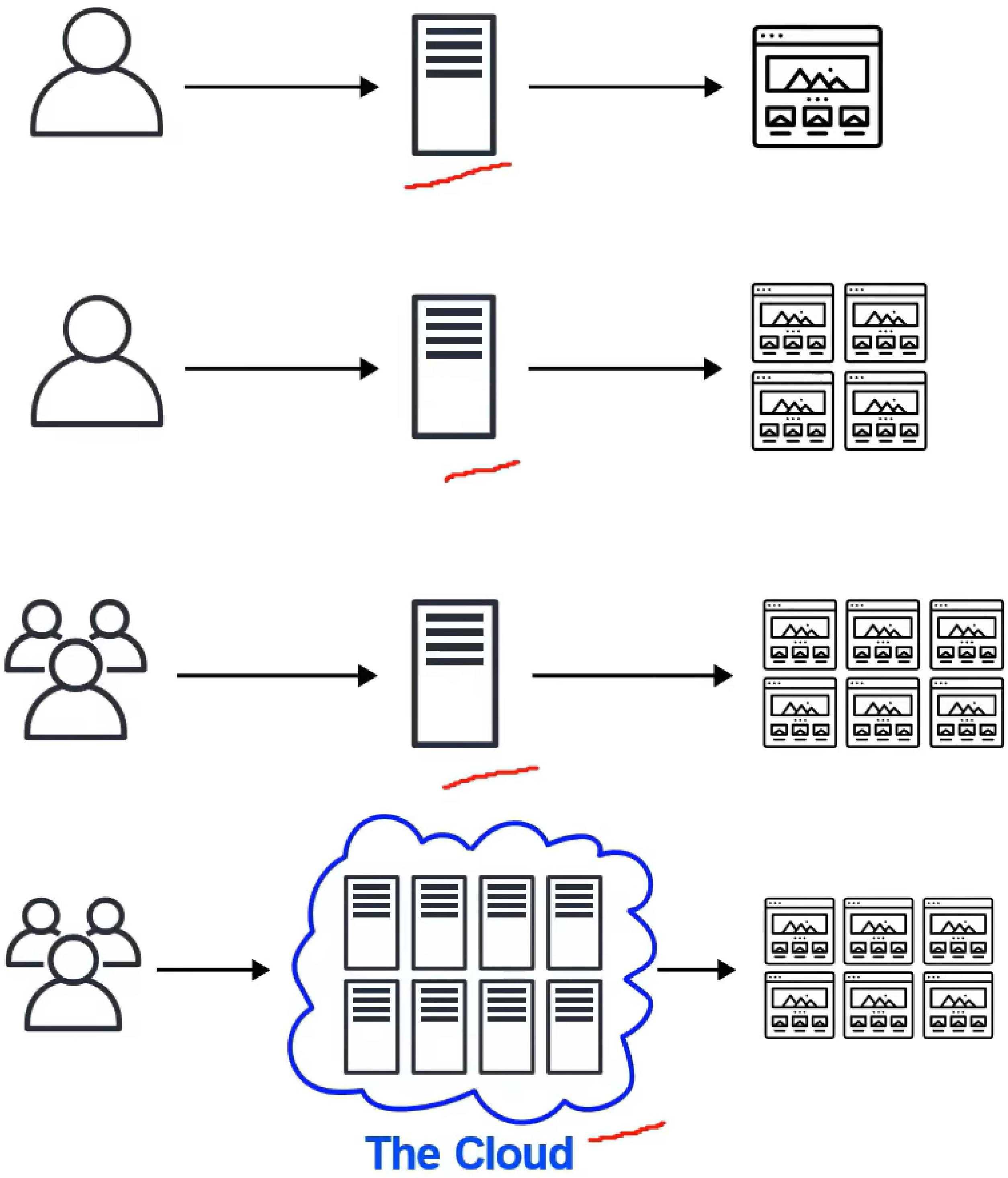
characteristics:
1. extremely cheap
2. LIMITED FUNCTIONALITY, BOTH BY THE MACHINE'S CAPACITY AND THE LACK OF FUNCTIONALITY OFFERED.
3. Poor isolation of resources : If one person decided to use a large portion of the resources, that would hang and hinder ALL websites on the server.

4. Cloud Hosting

MULTIPLE physical machines, that act as ONE system. (The computing is distributed)

The system is abstracted and broken down into multiple *cloud services*, and each Physical machine in the cloud centre runs a certain assigned number of services, making things much more efficient.

characteristics:
1. Flexible and scalable
2. Highly configurable
3. More cost effective and secure (depends)



Common Cloud Services

A cloud provider can have hundreds of service types, that can be group under some major categories.

4 most common ones for IaaS (Infrastructure as a Service) are:

1. Compute --> Virtual computers to run code and programs. (Wildly powerful systems)

2. Networking --> Virtual Networks, easily configure network isolations and virtual networking
3. Storage --> Self explanatory
4. Databases --> Virtual huge prebuilt/ most pre-configured Databases to store report data or gen purpose web-app databases, etc.

What is Azure?

Microsoft's Cloud service (similar to AWS by amazon, or Google Cloud.)

> Cloud Computing has now become the default umbrella term for all the services provided. So, even storage on the cloud is an example of "cloud computing."

Benefits of Cloud Computing

1. Cost Effective

1. No upfront cost
2. PAYG (Pay as you Go) model makes sure that that the clients pay for the resources they consume.

// However, despite this being what is said on the exam, in real life, this can get extremely costly as the organisation and its resource usage scales. It is therefore wise to know when to switch, and plan it from the start. There are many organisations where the amount of DB and storage resources that they use at a particular cloud provider is so much, that even transferring / downloading the data to their own servers is too exorbitant a cost, and then, the organisation is more or less forever bound to one certain provider.

2. Global Availability

Workloads can be launched anywhere in the world, by simply choosing the region. (this is because the storage and processing is done on the cloud, so the clients are not geolocation bound.)

3. Security

1. The physical Security is undoubtedly high and much better at huge corporation owned datacentres, than what a small org could muster up themselves.
2. Cloud services an be *secure by default* or the client can *configure* their own security settings as well.

4. Reliability

Lot of prebuilt, well maintained and configured solutions for Data Backup, Disaster Recovery, Fault tolerance and Data Replication.

From ChatGPT:
Here's a breakdown of the terms mentioned:

1. **Data Backup**:
   - **Definition**: Data backup involves creating copies of data to ensure it can be restored in case of data loss, corruption, or hardware failure.
   - **Purpose**: To protect against data loss due to accidental deletion, hardware failure, software issues, or disasters.
   - **How It Works**: Backups can be full (a complete copy of all data), incremental (only changes since the last backup), or differential (changes since the last full backup). These backups are typically stored in separate locations (offsite or cloud storage) to enhance security.
   2. **Data Recovery**:
   - **Definition**: Data recovery is the process of restoring data from backups after it has been lost or corrupted.
   - **Purpose**: To return to a known good state of data following a failure or issue.
   - **How It Works**: Depending on the backup strategy, data recovery might involve restoring the most recent backup or a specific version of data. Recovery plans should be tested regularly to ensure effectiveness.
   3. **Fault Tolerance**:
   - **Definition**: Fault tolerance is the ability of a system to continue operating properly in the event of a failure of some of its components.
   - **Purpose**: To minimize downtime and maintain service availability even when parts of the system fail.
   - **How It Works**: Fault tolerance can be achieved through redundancy, such as using multiple servers or network paths so that if one fails, another can take over. It can also involve using error-correcting codes or failover mechanisms to ensure continuous operation.

4. **Data Replication**:
- **Definition**: Data replication involves copying data from one location to another to ensure consistency and availability across different systems or sites.
- **Purpose**: To enhance data availability, support load balancing, and improve fault tolerance by keeping multiple copies of data.
- **How It Works**: Replication can be synchronous (where changes are made to all copies at the same time) or asynchronous (where changes are made to the primary copy first, then replicated to secondary copies). This process can be implemented across different geographical locations or within the same data center.

In summary:

- **Backups** are about making copies to recover from data loss.
- **Recovery** is about restoring those copies when needed.
- **Fault Tolerance** is about ensuring the system keeps running even if parts fail.
- **Replication** is about keeping copies of data in multiple places to improve availability and performance.

5. Scalability

As the organisation's need for resources increases, simply buy more computing power, without any additional infrastructure.
Increase/Decrease resources/services acc. to demand

6. Elastic

Automation of scaling options (increasing resource pool when there is a spike in need, and decreasing when there is little need) makes pricing more economical for the clients.

7. Current (updates and patching)

The Cloud provider does the updating, security patching and all other relevant configurations on their machines without any interruption (in most cases) to the client and their services.
The Cloud provider distributes services to different systems and load-balances traffic on their own, to ensure seamless resource availability for the client.

Types of Cloud Computing

1. SaaS

Software as a Service

Product is run and managed by the provider. (intended for customers)

We do not have to worry about how the service is maintained, it just works and remains available for us.

ex: Office 365, Gmail

2. PaaS

Platform as a Service

Focus for the user is on building, development and management of their apps. (primarily useful for developers)

We do not have to worry about provisioning, configuring or understanding the hardware or OS.

ex: ElasticBeanStalk(AWS), heroku, Google app engine

3. IaaS

Infrastructure as a Service

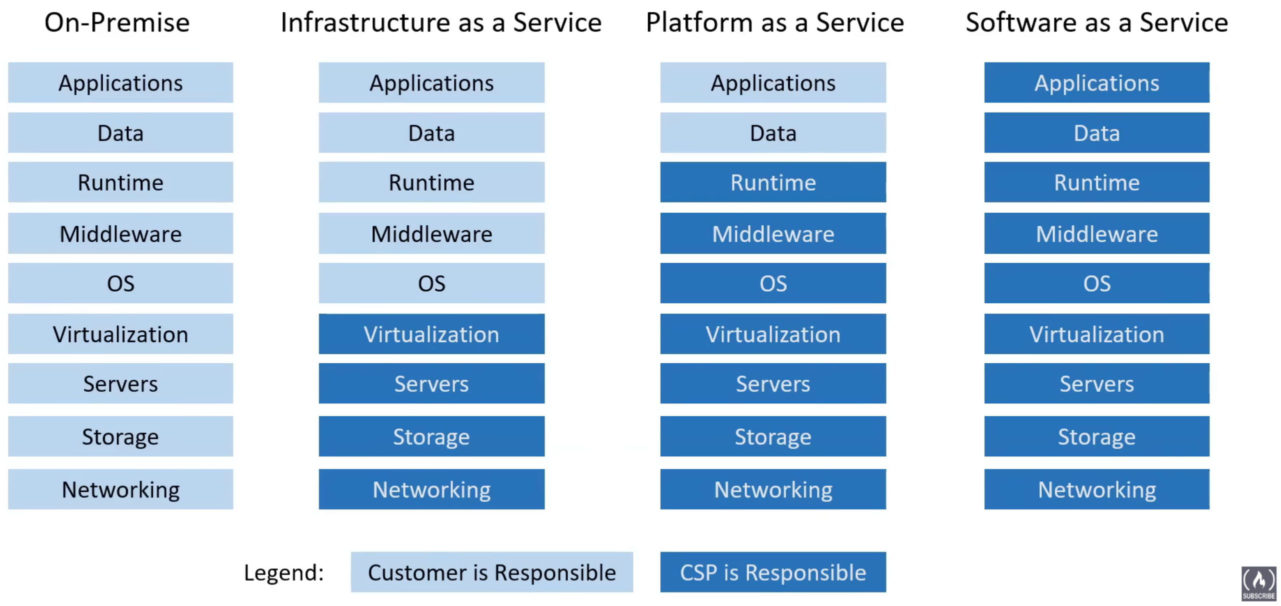
Basic building blocks for cloud based IT. Networking features, computer controls and data storage is available (useful for SysAdmins)

We do not have to worry about IT staff, hardware and data centres.

ex: Azure, AWS, Oracle Cloud , google Cloud

> A lower strata can still have the other services built and provided on top of it.

Types of Cloud Computing Responsibilities



On-Premise	Infrastructure as a Service	Platform as a Service	Software as a Service
Applications	Applications	Applications	Applications
Data	Data	Data	Data
Runtime	Runtime	Runtime	Runtime
Middleware	Middleware	Middleware	Middleware
OS	OS	OS	OS
Virtualization	Virtualization	Virtualization	Virtualization
Servers	Servers	Servers	Servers
Storage	Storage	Storage	Storage
Networking	Networking	Networking	Networking

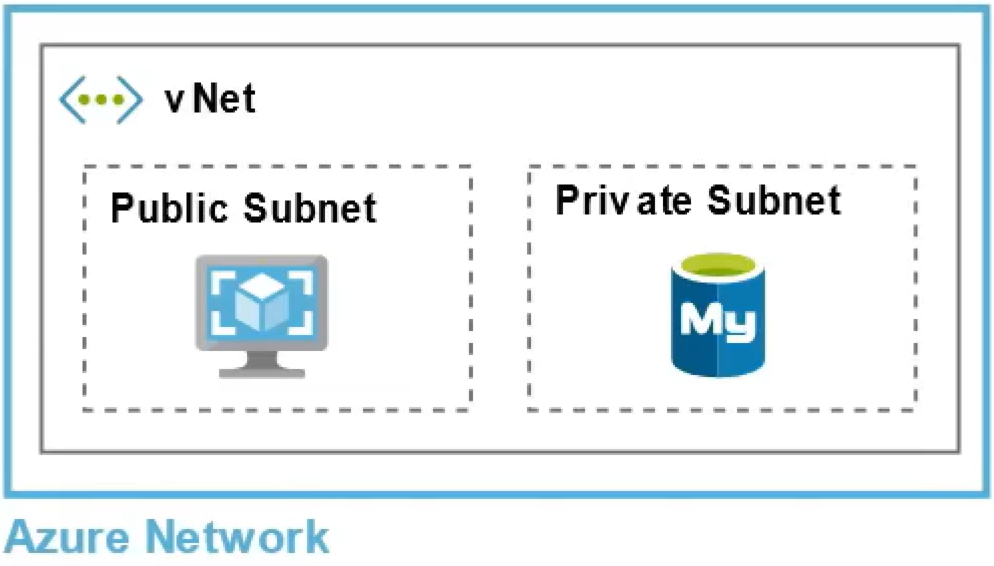
Legend: Customer is Responsible CSP is Responsible

// On premise --> Private cloud, owned and maintained by the organisation themselves.

Azure's Deployment Models

1. Public Cloud (Cloud-Native)

EVERYTHING is built using the Cloud Service Provider.

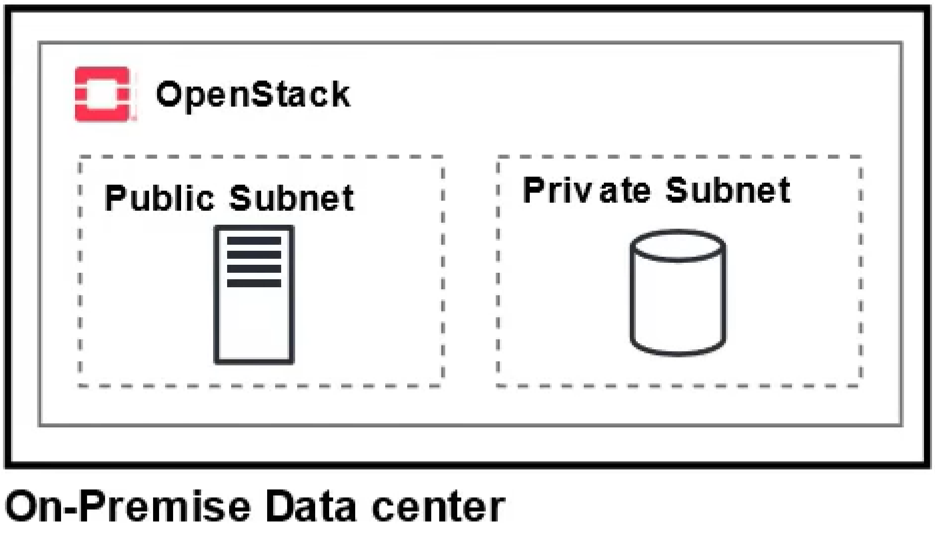


// An Azure network with the virtual machine and database running on it, as an architectural diagram example.

2. Private Cloud (On-Premise)

EVERYTHING is built on the organisation's own datacentres.

Could be a private cloud design and construction, or, it could be an Open Source cloud design/ software (like, OpenStack).



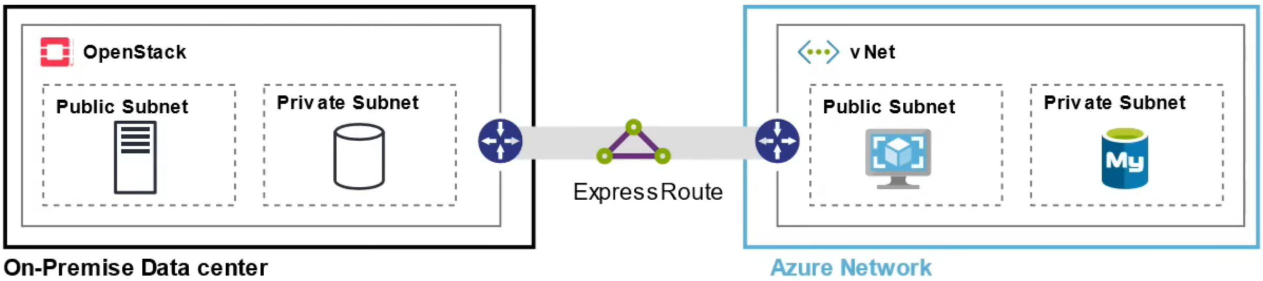
// Diagrammatic representation of the OpenStack using Private-cloud deployment model.

3. Hybrid

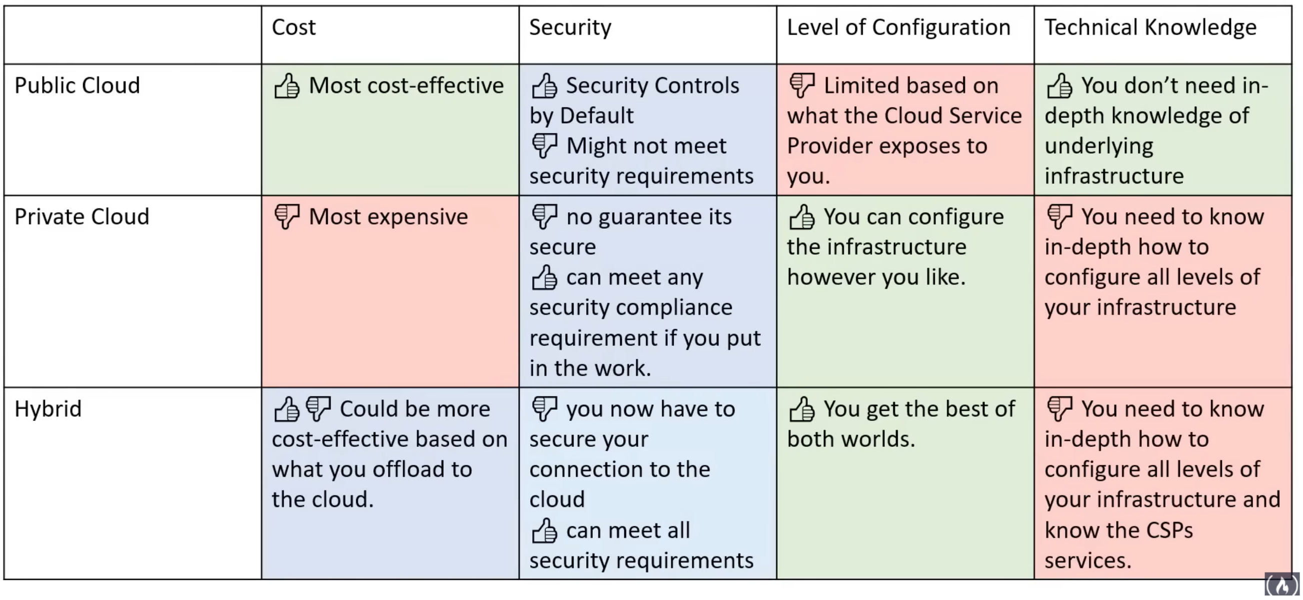
Uses BOTH On-premise resources and the CSP's resources.

Both are connected; softwares like ExpressRoute can help to accomplish that.

> ExpressRoute is a dedicated connector service, which is logically like a fibreoptic line running b/w the private and CSP resources.



// On-premise (private) and Cloud-native resources connected together using ExpressRoute in this case.

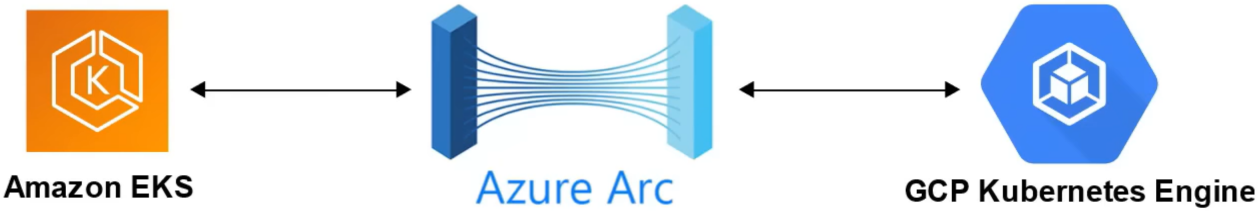


	Cost	Security	Level of Configuration	Technical Knowledge
Public Cloud	👍 Most cost-effective	👍 Security Controls by Default 👎 Might not meet security requirements	👎 Limited based on what the Cloud Service Provider exposes to you.	👍 You don't need in-depth knowledge of underlying infrastructure
Private Cloud	👎 Most expensive	👎 no guarantee its secure 👍 can meet any security compliance requirement if you put in the work.	👍 You can configure the infrastructure however you like.	👎 You need to know in-depth how to configure all levels of your infrastructure
Hybrid	👍👎 Could be more cost-effective based on what you offload to the cloud.	👎 you now have to secure your connection to the cloud 👍 can meet all security requirements	👍 You get the best of both worlds.	👎 You need to know in-depth how to configure all levels of your infrastructure and know the CSPs services.

4. Cross-Cloud (Multi-Cloud/Hybrid-Cloud)

Using Multiple cloud Providers for multiple services.

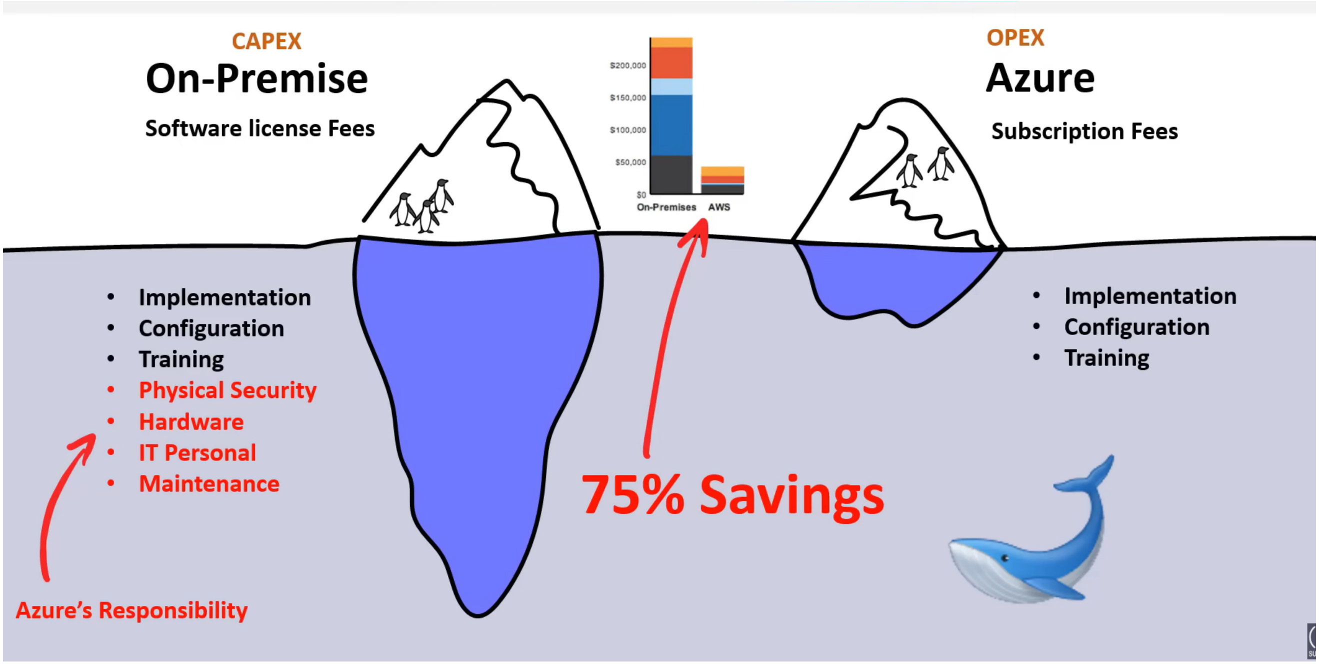
And all of them will be treated as if they are on the same network.



// Virtual Machines and containers running on different CSP services are treated as if they are on the same network.

TCO (Total Cost of Ownership)

Comparing the Costs of different models, with both upfront, apparent costs and hidden ones.



CAPEX
On-Premise
Software license Fees

OPEX
Azure
Subscription Fees

- Implementation
- Configuration
- Training
- Physical Security
- Hardware
- IT Personal
- Maintenance

Azure's Responsibility

75% Savings

- Implementation
- Configuration
- Training

// CAPEX --> Capital Expenditure

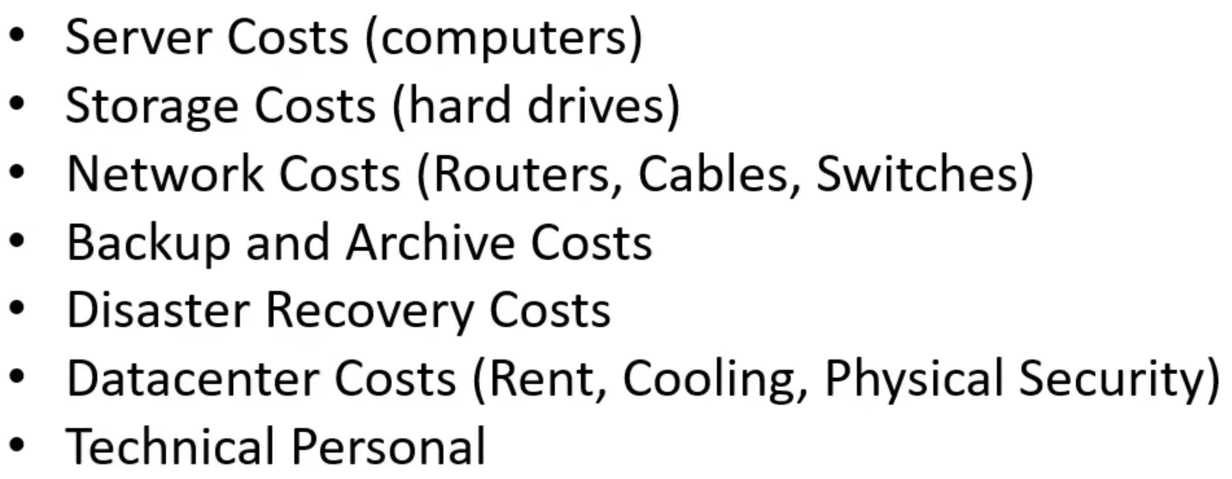
// OPEX --> Operational Expenditure

CAPEX vs OPEX

Capex

Spending money upfront on physical infrastructure.
Deducting that expense from your tax bill over time.

Some things that would be considered a Capital Expenditure:



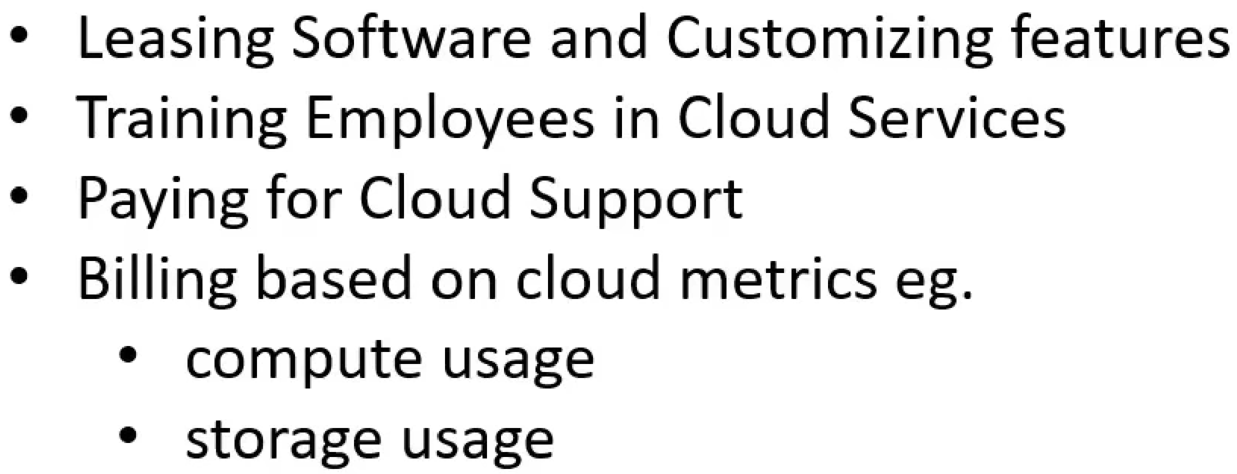
- Server Costs (computers)
- Storage Costs (hard drives)
- Network Costs (Routers, Cables, Switches)
- Backup and Archive Costs
- Disaster Recovery Costs
- Datacenter Costs (Rent, Cooling, Physical Security)
- Technical Personal

// With CAPEX the organisation has to *guess* up-front, what it plans to spend

OPEX

The on-premises and physical hardware related costs are handled by the CSP, and the client org only has to be concerned with the non-physical costs.

Non-physical costs:



- Leasing Software and Customizing features
- Training Employees in Cloud Services
- Paying for Cloud Support
- Billing based on cloud metrics eg.
  - compute usage
  - storage usage

One huge advantage is that services and products can be tried and check for the org's personal needs without any investment in equipment (which is very hefty, usually.)
Huge reduction in Costs and flexibility are the main attractions.

Cloud Architecture Terminologies

Solutions Architect

A technical role, wherein the architect devises and designs a technical solution using multiple systems via:

1. Research
2. Documentation
3. Experimentation

Cloud Architect

A Solutions architect that is focused solely on architecting technical solutions using Cloud Services.

// Practically speaking, "Solutions Architect" is often used to refer to both cloud architect and the original solutions architect.

Terms that a Cloud Architect must understand and factor into their designs:

1. Availability : Ability to ensure that a service remains available (HA high availability is best)
2. Scalability : Ability to grow rapidly, unimpeded, along with the demands of the business
3. Elasticity : Ability to shrink/grow/adjust resource usage to meet the requirements
4. Fault Tolerance : Ability to prevent failures in the event that faults occur
5. Disaster Recovery : Ability to retrieve data and resume operations smoothly in the event of a loss. Ability to recover form a failure. (DR Highly Durable infrastructure design is the best)

A solutions Architect must consider the following *business factors* as well:

1. Security --> *How secure a solution is. Security must be according to the needs of the operations, slightly higher is better. However, there is no point in deploying Military grade security for a service that is , say, a Newsletter/blog. Considering that and picking the appropriate security solution is also the part of the Solutions Architect's job.*
2. Cost --> *How much does the solution cost? And deciding the right balance b/w performance compromise and cost efficiency to ensure smooth operations for the organisation.*

Cloud Architect Terms in some Detail

High Availability

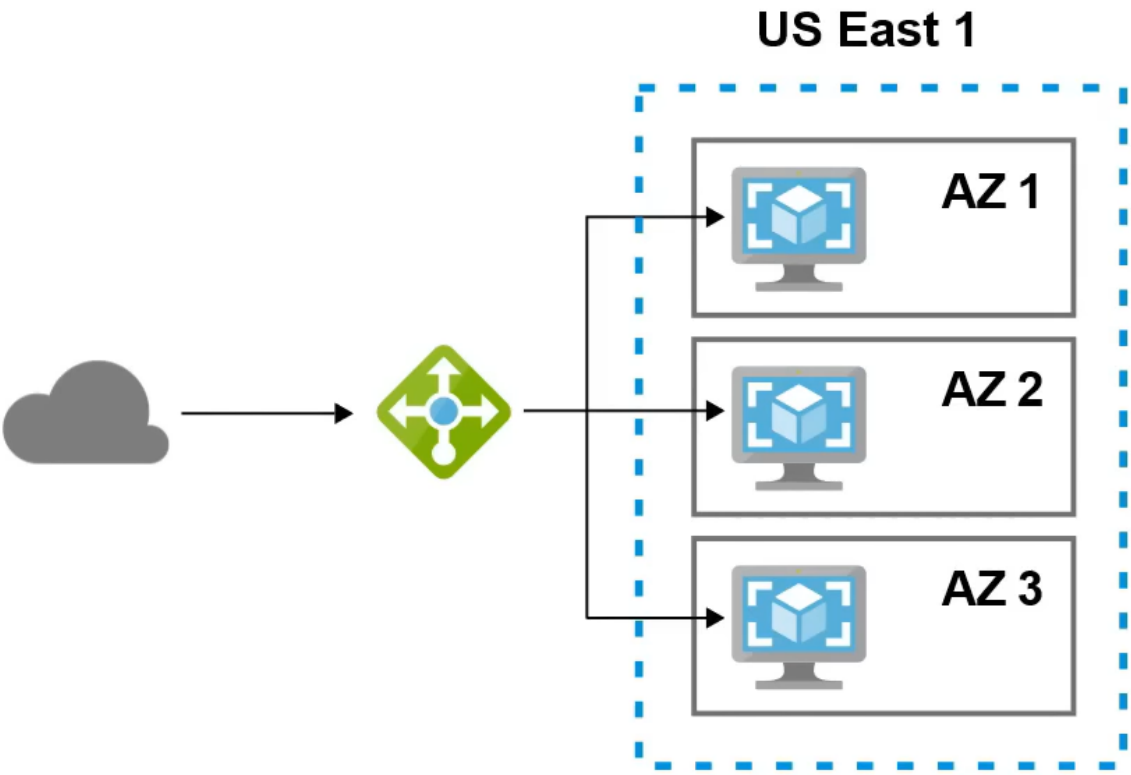
The ability of the service to remain available by ensuring that there is no *single failure point* and/or ensure a certain level of performance.

What the "single point of failure" means in this context is that, for example, all the services and their data were concentrated at one singular datacentre, and something happened that caused a lack of connection to the datacentre, this would mean nothing is available anymore.
Here the connection to the Datacentre is a potential single point of failure.

> Having Multiple Servers at Multiple Datacentres is great.

Availability Zones: What Azure calls their Datacentres.

So, running and distributing the organisation's workload b/w multiple Availability Zones ensures the Availability of the application/service even in the event that one or two AZs are down.



// The green icon is the Azure Load Balancer.

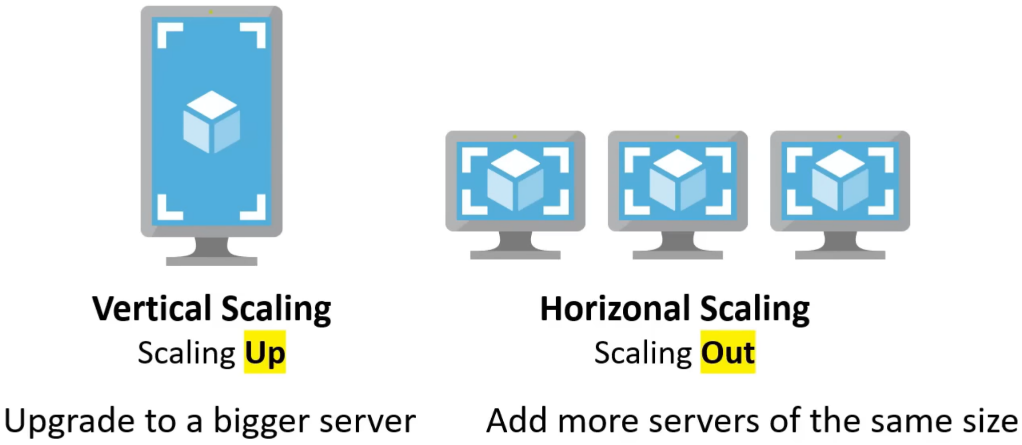
How does this process of distributing the load across multiple AZs work?
It works with the help of the Load Balancer. Analogous to LoadBalancing often seen in Networking.

Azure Load Balancer
Allows the client to evenly distribute the traffic to multiple servers in one or more AZs.
If one AZ becomes unavailable, Load Balancer will route the traffic to the available AZ servers.

High Scalability

The ability of an organisation to increase their capacity acc to the increasing resource demands.

1. Vertical Scaling : Upgrading to bigger , more powerful server machines (AKA Scaling Up)
2. Horizontal Scaling : Upgrading to multiple, similarly powerful servers and load balancing, to give the same results as a Scale Up. (AKA Scaling Out)



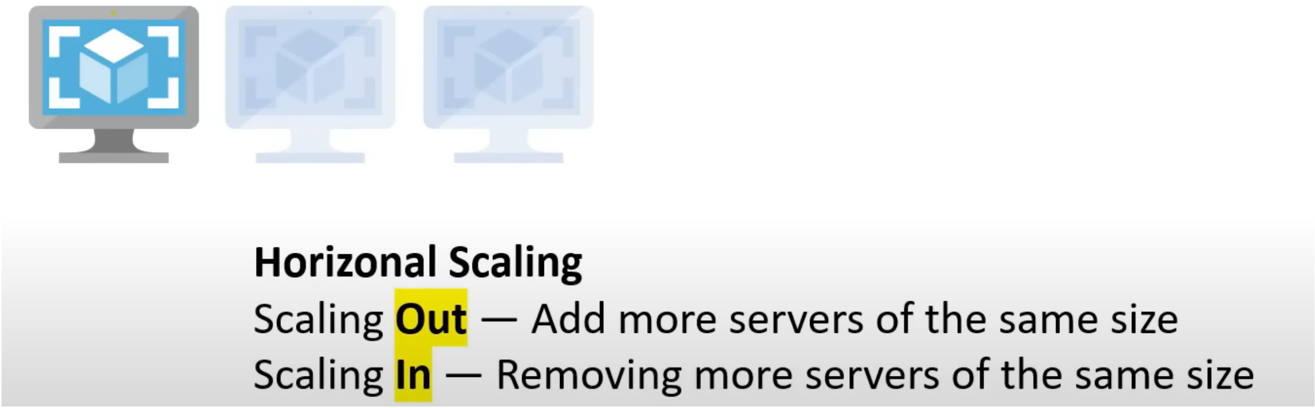
High Elasticity

Ability of an organisation to **AUTOMATICALLY** increase/decrease capacity and resources based on current needs.

Generally, Horizontal scaling is used with Elastic Architectures.

Scaling out --> Adding more servers of similar size
Scaling in --> Removing servers of similar size

> Vertical scaling is generally reserved for using with traditional architectures. Because data loss upon upgradation is common.


Horizonal Scaling
Scaling Out — Add more servers of the same size
Scaling In — Removing more servers of the same size

How to implement Elasticity in our architecture?

1. Azure VM scale sets --> Automatically inc/dec resources in response to demand fluctuations OR defined schedule //more on this later.

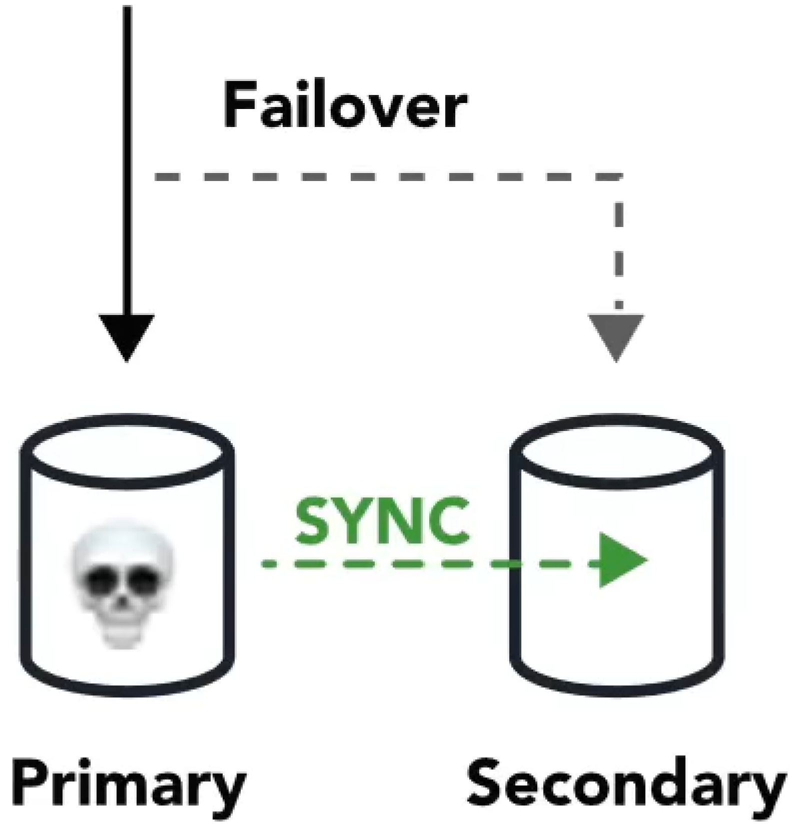
2. SQL Server Stretch DB --> Dynamically stretch warm and cold transactional data from Microsoft SQL server 2016 to Azure

Fault Tolerance

The ability of an organisation to ensure that there is NO SINGLE POINT OF FAILURE, significantly reducing and potentially preventing chances of failure.

Fail-Overs

When there is a plan to shift traffic to a redundant system in case of failure of the primary system.



// A common example is having secondary (replica) of your database where all ongoing changes are synced.
// This DB comes into use only when the primary DB fails, a fail-over occurs and this secondary one, mostly updated, takes over smoothly, and is promoted to the Primary Role.

Azure Traffic Manager

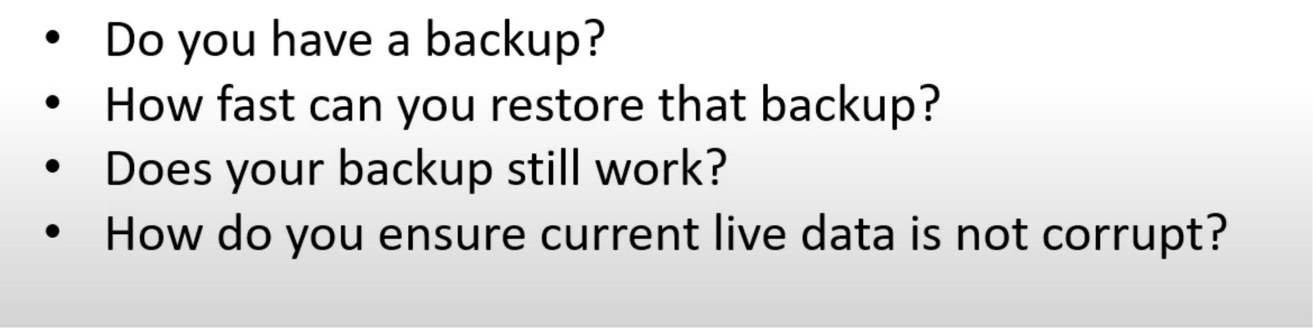
A DNS based Traffic balancer to fail-over from a malfunctioning Primary system to a On-standby secondary one. (A loadbalancer can do this as well)

High Durability

Ability to recover from a Disaster and PREVENT THE LOSS OF DATA in such a scenario.
Solutions that help in this are DR (Disaster Recovery) methods and solutions.

Architectures with robust DR systems are Highly Durable.

Example of DR questions to ask before implementation:

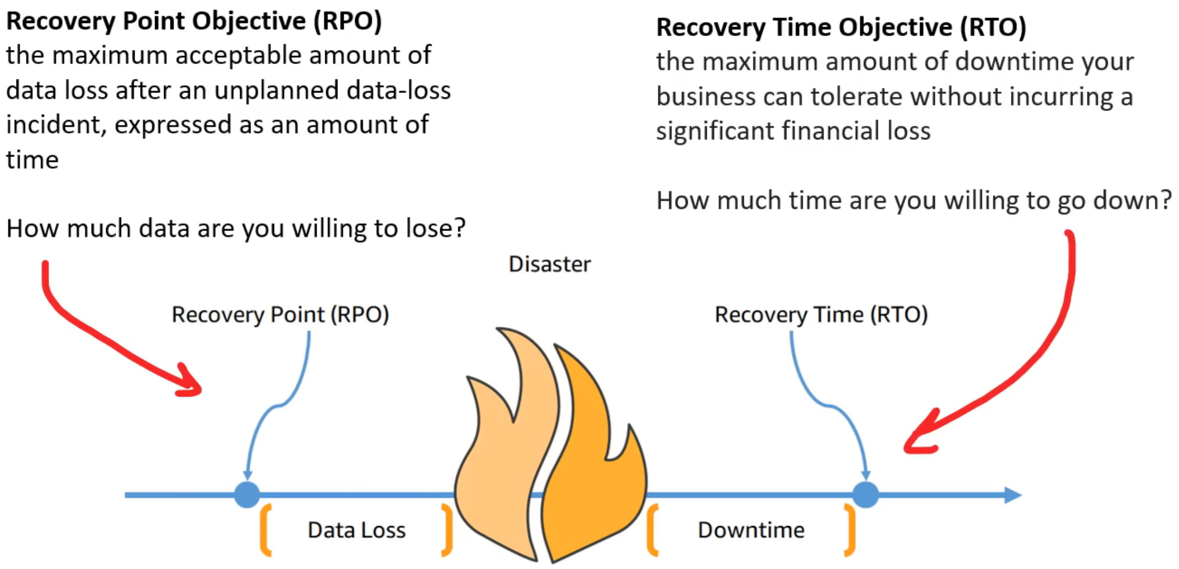


- Do you have a backup?
- How fast can you restore that backup?
- Does your backup still work?
- How do you ensure current live data is not corrupt?

BCP (Business Continuity Plan)

A BCP is a document that outlines how a business will continue operating in the event of a hitch or unplanned disruption in the services.

Might include Factors like RPO and RTO, as in this example:



Recovery Point Objective (RPO)
the maximum acceptable amount of data loss after an unplanned data-loss incident, expressed as an amount of time

How much data are you willing to lose?

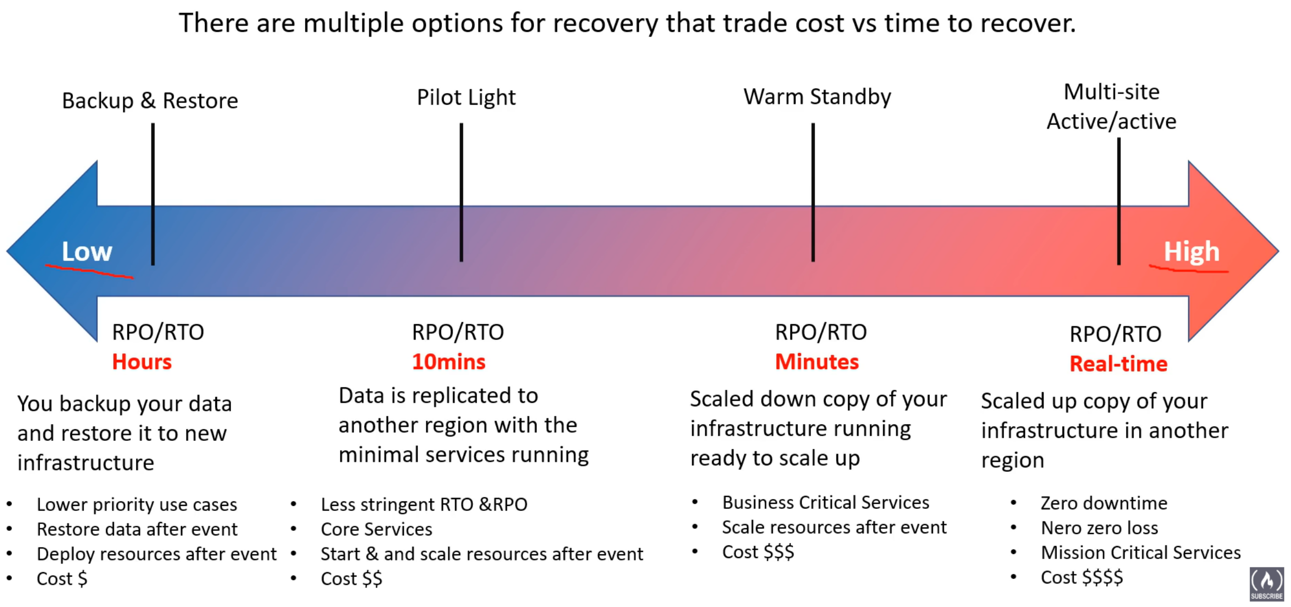
Recovery Time Objective (RTO)
the maximum amount of downtime your business can tolerate without incurring a significant financial loss

How much time are you willing to go down?

Disaster Recovery Options

Multiple options are available for Disaster Recovery depending upon the trade-offs that the organisation is willing to make. { RPOs VS RTOs (Recovery Point Objectives vs Recovery Time Objectives). }

Usually the lesser data the org is willing to lose, the more downtime it takes while recovering. And vice versa. A good fit is the balance achieved according to what the organisation truly needs.



There are multiple options for recovery that trade cost vs time to recover.

Backup & Restore

RPO/RTO
Hours

You backup your data and restore it to new infrastructure

- Lower priority use cases
- Restore data after event
- Deploy resources after event
- Cost $

Pilot Light

RPO/RTO
10mins

Data is replicated to another region with the minimal services running

- Less stringent RTO &RPO
- Core Services
- Start & and scale resources after event
- Cost $$

Warm Standby

RPO/RTO
Minutes

Scaled down copy of your infrastructure running ready to scale up

- Business Critical Services
- Scale resources after event
- Cost $$$

Multi-site Active/active

RPO/RTO
Real-time

Scaled up copy of your infrastructure in another region

- Zero downtime
- Nero zero loss
- Mission Critical Services
- Cost $$$$

Recovery Times*:

1. With backup and restore (wherein, it is only raw data that is saved) --> Hours/ Days of Recovery Time
2. With Pilot Light --> 10-20 Minutes
3. Warm Standby --> Minutes
4. Hot Standby/ Active-active multi site --> None, seconds/ Realtime