# Cloud Computing Intro

**Eli the Computer Guy - November 4, 2024**

## What is the "Cloud"?

The "cloud" referred to the Internet when tech professionals were diagramming their networks. They'd have a detailed layout of their network, and then their modem would simply be connected to "the cloud".

This reference was used because Internet networking equipment continuously finds new routes for destinations so that no two connections between networks will be the same.

## What is "Cloud" Computing?

20 years ago if your company needed email services you would have a physical server, with an operating system installed and Exchange Server installed to route email. If "email was down" you could point to the exact physical server and know that it was where the problem was.

"Cloud" Computing means if a service fails you kinda wave your hand in a direction and say "something in there failed". Cloud computing means the service resides in a "cloud" of systems. You may have a rack of 20 physical servers that provide availability to "Virtual Server Instances". The Email Virtual Server might be running on Physical Server 1, or 10. You just point to the rack.

When you use AWS or Azure this idea of "cloud" becomes more important as you literally have no idea where the datacenter that hosts your Virtual Machines may be.

With Cloud computing we focus more on logical design or systems then physical design.

## Abstraction

Abstraction is the concept of turning logical subsystems into their own architecture. Instead of an OS installed on a hard drive or a physical computer with a specific CPU you "abstract" out the resources.

This means you select a Compute level appropriate for your project. For an AI system you may need an expensive NVIDIA GPU Compute unit, but for a file server you may need a slow one. Storage is a separate system such as a SAN (Storage Area Network) locally, or one of the many options AWS and Azure have.

- Compute
- Storage
- Database

## Services and Micro Services Architectures.

Every function a server has includes a number of vulnerabilities. The more functions a single server has the easier it it to be hacked. Now that cloud computing makes it cheaper to create single function servers you can have 10 virtual servers to provides the services a single server used to provide. So instead of a Windows Small Business Server that would offer AD, File,

VPN, Email, etc services on one box you can now have individual instances offering a single service.

# Colocation Facilities

Colocation facilities are where you rent racks or rack space in a vendors datacenter.  This gives you the power, environment, and bandwidth resources of a full datacenter while only paying for what you use. Even if you decide to use your own hardware and manage systems yourself it may make sense to house your systems in a colo facility.

# Virtualization and Hypervisors

Virtualization is the process of installing an Operating System into a "Virtual Instance". Instead of the OS being installed onto hardware, you install the OS into a special type of file.  A hypervisor is the installed onto the physical hardware and it can run virtual instances.  This means a full server is nothing more than a file that can be moved relatively easily between hardware.

## Failover and Load Balancing

When you design an architecture based around virtual machines you can easily move Instances between physical servers easily. You may do this for power management.  Servers that are under utilized will be brought onto a single server, and then as load increases they can be moved to other servers as power is needed.

You can also have "failover" and more importantly "high availability failover".  This means is a CPU fan dies on a physical server and the hardware crashes the instance will be automatically moved to another piece of hardware and users should not notice any down time.  Entire physical servers become as swappable as hard drives in a RAID.

### Type 1 Hypervisor (Bare Metal)

For this form of virtualization a hypervisor is installed onto the physical server.  This is a lightweight OS that is only used to manage Virtual Machines.

You then use a Management system to connect to your hypervisor systems and install Virtual Machines and configure how they work.

For clusters of hypervisors you many times will store the virtual machine files on a SAN so that the hypervisor machines are only running the Instances that are stored on the SAN.

**WARNING**: Sometimes vendors offer the hypervisor as a free open source product, but the management software is very expensive.

### Type 2 Hypervisor (Virtualbox)

A type 2 hypervisor is a piece of software installed onto a normal computer.  You then use it to create and manage virtual machines.  This is good for testing, and when you may need tools that your primary OS does not support.

## Hypervisor Configurations

When you setup a virtual machine on a hypervisor you will need to state what resources it needs.  Modern hypervisors allow this to be somewhat dynamic, but you can run into configuration bottlenecks.

If you have a system with 16GB of RAM and you plan to run VM's on it you will need to determine how much RAM the host machine needs.  Then figure out how much each instance you plan to run will need. If you over allocate RAM or other resources the whole setup can become "quirky".

# Containers

Containers are a form of virtualization that allows you to silo environments on a server.

# aaS - as a Service

The "as a Service" model means that you only maintain the resources that there is a need for you to maintain. Outlook used to be installed on every computer and would connect to an Exchange Server that would have to be installed and maintained.  Now you simply pay for Office 365 or Google Apps and the end user gets the same results. The administrative overhead has moved from your IT department to a vendor.

## aaS Cost

The cost for a monthly subscription can generally be deducted in whole from taxes by the company instead of having a depreciation schedule where the company can only deduct a certain amount per year for equipment they own.

Support staff and consultants are expensive.  1000 users on an Office 365 plan for $10 a month is "only" $10K per month and $120K per year.  Add up an employee's salary, benefits, and the cost of hardware and things like disaster recovery products and aaS many times is a cheaper solution based on TCO.

## aaS Security

yup... it's a concern.

## Types of aaS

SaaS - Software as a Service

This is a full fledged product such as Office 365, Adobe Online, or Salesforce

PaaS - Platform as a Service

This is where the basic platform is subscribed to and you simply upload your code and have little ability to change configurations.  This is like web hosting services where they allow you to upload HTML and PHP. You are only responsible for your code

### IaaS - Infrastructure as a Service

This is where core infrastructure components are in the cloud.  Database servers, Storage, Compute, and even networking services are simply a resource you pay a fee to access.  This is like AWS or Azure.

### MaaS - Metal as a Service

This is where you are able to install an OS on to a "bare metal" server.  Just like you are fully responsible for the server at your facility you are responsible for almost everything on this server.  From OS updates and Patches to configuration management.

This used to be a better option 10 years ago before virtualization infrastructure had matured.

# AWS and Azure

AWS and Azure are IaaS providers. You design and configure how yo want you architecture to logically work, and then purchase the services that allow you to build that.

These offerings have become incredibly complex and you really need to know your requirements and budget when you design your system.

Choosing between AWS and Azure is generally based on your tech stack.  Azure is maintained by Microsoft and so is far more integrated into that stack than AWS.

# Digital Ocean and Linode

DigitalOcean and Linode provide a more scaled back IaaS experience.  AWS and Azure offer so many services and variations of services it is a legitimate problem trying to figure out what a small project may need.  With Digital Ocean and Linode they have fewer options and a streamlined experience.

# Clustering

Clustering is generally used for database systems and allows you to connect multiple database servers into a single cluster. The cluster adds fault tolerance and load balancing for your systems. If a single server in the cluster fails it can simply drop off line without the users noticing.  For performance user requests will be balanced over the systems in the cluster so no single server is overloaded by requests.

# Application Servers

These are sorta like a modern mainframe.  A server computer runs installed software within virtualized environments and then users use a "thin client" to connect to the application server and use the software.

This may be used when legacy software needs to be accessed remotely. 10+ years ago accountants would use application server vendors to allow cloud based access to different versions of Quickbooks.

# Serverless Architecture

Serverless Architecture allows you to use compute resources on another server.  This can be things like Vision, Voice or AI services where you provide data, the system processes it, and you get results back.

**Example:** Azure Cognitive Services - AI Like Systems

"Serverless" does not mean servers are not used anywhere. It simply means you are not responsible for the maintenance and upkeep of the server system.

# Reverse Proxy Servers

Reverse Proxy servers act as load balancers to send users to servers in a cluster.  You may spin up 20 servers for your platforms front end.  Users will then be auto directed to the most appropriate server.

# Geo DNS

Geo DNS allows you to provide different IP Address resolution based on where the visitor is connecting from. This can be used for latency, or regulatory purposes so that when a visitor goes to "your" platform they are actually directed to the infrastructure that is most appropriate for them.

# Caching and CDN's

Caching allows you to dynamically store content pulled from server systems into storage you control.  This allows for access to the content without having to make requests from the main repository.

CDN's - Content Delivery Network - act as a distributed caching mechanism for visitors of your platform.

You create caching infrastructure physically close to your expected visitors.  When the first visitor from an area requests a piece of content that content is pulled from the main repository and sent to the client and to a cache in the CDN.  When the next visitor requests the same resource the content is pulled from the CDN instead of the main repository.

# Public, Private and Hybrid Clouds

"Cloud" simply means you can't point to the exact "box" that is doing something.  You can create a cloud with your own hardware that is hosted in racks you own, you can build your systems online, or a mixture of the two.

## Private Cloud

A private cloud is one that you own.  A rack of Type 1 Hypervisor Servers that connect to a SAN.

## Public Cloud

Public Clouds are owned by vendors such as AWS or Digital Ocean.

## Hybrid Clouds

Hybrid Clouds are infrastructures where the Private and Public Clouds are logically connected together.  Such as you may have a local MSSQL server at your site that replicates to an Azure SQL Server.  This gives you local performance with real time backup if something happens to your physical server.

Or you may have virtual servers running locally that you send up to a vendor if load increases.  Such a server for the marketing department for gaining customers.  Normally the server is under light load, but the company paid for a Super Bowl ad.  The server can be moved to a higher capacity infrastructure provided by a vendor while the ad viewers some to your site, and then be pulled back once the campaign is over.

# Orchestration Software

Orchestration systems allow you to easily administer a large number of servers and systems.  Puppet, Chef, Ansible, OpenTofu allow you create, manage and maintain your environment with a 'infrastructure as code" mindset.

# Vendor Stacks and Lock In

Picking a vendor generally depends on the tech stack you are using.  If you are already married to Windows then Azure is probably the best bet.  If you need a large number of services related to a linux environment then AWS may be the way to go.

Lock in is a real concern with using cloud vendors.  It is not simply about data being "portable" but rather the form you can export you data in will be something you can easily deal with.  Additionally inbound network traffic is generally free or cheap, but outbound can be pretty expensive.

# Is the Cloud Secure?

How much do you really trust your employees or yourself? Any system connected to the Internet is on the "cloud".

## Disaster Recovery in the Cloud

IaaS providers warn that regions can go down.  Creating failover within an IaaS infrastructure may make sense if it makes financial sense.

**Question:**
• What happens if your aaS provider goes bankrupt?
• Is your backup provider simply using the other vendor on the backend?

## DRaaS - Disaster Recovery as a Service

Since we abstract the components of our infrastructure into movable parts there are vendors that can provide Disaster Recovery Services that are far better then what could be previously provided.

## Multi Cloud Architecture

**Question:** But really...???

For reliability you can have your infrastructure span multiple public clouds. An AWS database could replicate to an Azure service.

This is expensive and complicated to maintain.

# Regulations

Your job goes beyond configuration files...

## PCI, HIPPA,GDPR

Vendors will tell you if they comply with specific regulations and laws.  It is up to you to determine if you need to follow these types of rules, and then find the appropriate vendor or product.

## Data Sovereignty

Some Countries require that data about their citizens must physically be stored in their countries

# TCO and Architecture

Total Cost of Ownership has to be factored in when designing your infrastructure. A cheap system to setup may be exorbitantly expensive to run at scale.

Tech Professionals generally cost $100K and up so paying for an "expensive" SaaS subscription may be cheaper.

Costs for cloud resources are cheap or expensive based on use and resource.  Compute and Databases are generally pretty cheap.  Data Storage and Bandwidth can be expensive for large volumes.  If you were going to build a YouTube clone it would make sense to build the Interface, user accounts, and data tracking systems on AWS or Azure, but then connect those to a Storage Server or SAN in a colocation facility.  The storage server will be a large upfront cost, but then a known ongoing cost.