Lecture: By Daniel James Higgins
Presentation: By Daniel James Films

Cloud Computing for Business, Machine Learning, and Artificial Intelligence

Introduction:
Cloud Computing - Data Storage - Coding Languages - Programming - Software Development - Machine Learning (Analytics, Algorithms) - Artificial Intelligence

Architecture, Infrastructure and Security for each element

Introduction to Cloud Computing

INTRODUCTION

Part 1
Definition and Essential Characteristics of Cloud Computing
Delivery of on demand computer resources via the internet

CLOUD COMPUTING (Analytics) - definition
A model for enabling convenient, on demand network access to a shared pool of configurable computing resource that can be rapidly provisioned and released with minimal management effort or service provider effort.
NETWORKS - SERVERS - STORAGE - APPLICATIONS - SERVICES

5 Essential characteristics, 3 deployment models, 3 service models

5 Essential characteristics

- On demand self service
    - Access to cloud resource, processing power, storage, network, using a simple interface without requiring human interaction with each service provider
- Broad Network Access
    - Cloud computing resources can be accessed via network through standard mechanisms and platforms (mobile phones, tablets, laptops, and workstations)
- Resource Pooling

    - Utilizes economies of scale, (bulk sharing)
- Rapid Elasticity
    - Access to resources is scalable (increases / decreases) when necessary
- Measured Service
    - Only pay for what you use or reserve as you go

3 Deployment Models

- Public
    - Leverage cloud services over the open internet on hardware owned by the cloud provider, but its usage is shared by other companies
- Private
    - The cloud infrastructure is provisioned for exclusive use by a single organization, it could run on-premises or it could be owned managed, and operated by a service provider
- Hybrid (public and private)
    - Mix of both public and private clouds

3 Service Models

- Infrastructure
    - IAAS - infrastructure as a service
    - Access to infrastructure and physical computing resources such as servers, networking, storage, and data center space, without need to manage or operate them.
- Platform
    - PAAS - platform as a service
    - Access to the platform, that is hardware and software tools, usually needed to develop and deploy applications to users over the internet
- Application
    - SAAS - software as a service
    - Software licensing and delivery model, which software and application are centrally hosted and licensed on a subscription basis, referred to sometimes as “on-demand software”

Cloud Computing (Evolution, History)

- 1950’s “mainframes” - multiple users could access the same data storage
- 1970’s VM “virtual machines” allowed multiple mainframes - virtualization
- CLOUD COMPUTING IS BORN:
- “Servers” - created virtualization - multiple virtual machines connected and separated via “Hypervisor” (allows maintenance of individual virtual machines)
- “Pay as you go” for companies operating on cloud - switch from “CapEX” to “OpEx” (pay per usage based upon workloads as opposed to single up front cost for companies)

Key Considerations for Cloud Computing

Agility, Flexibility, Competitiveness --- of using the cloud

Infrastructure and Workload

- The cost of building and operating data centers can become astronomical
- Low initial costs and pay-as-you-go attributes of cloud computing can add up to significant cost savings

SaaS and Development and Platforms

- Paying for application access vs. buying software

Speed / Productivity

- Up and running on cloud vs. setting up traditional platforms
- Individual employee increases in productivity from using cloud resources

Risk Exposure

- Impact of cloud exposure, cloud pay as you go vs purchasing hardware solutions without trial of equipment

Benefits of Cloud Adoption

- Flexibility - scalability, customized applications, accessibility
- Security - virtual private clouds, encryption, API keys (keeps data secure)
- Efficiency - direct to market, accessibility, avoids hardware failure
- Strategic value - outsources solutions

Challenges of Cloud Adoption

- Data Security - business disruption
- Governance, sovereignty issues
- Legal, regulatory, compliance issues
- Lack of standardization of technologies
- Choosing the right service models
- Business continuity, disaster recovery

Key Cloud Service Providers and Their Services

- **Cloud Service Market/Models**
  - **IAAS** - **infrastructure as a service**
  - **PAAS** - **platform as a service**
  - **SAAS** - **software as a service**
- Cloud Service Providers
  - IMB Cloud
  - Google Cloud
  - Microsoft Azure
  - Amazon WEb Services
  - Oracle Cloud
  - Salesforce
  - Alibaba Cloud
  - SAP
- **Alibab Cloud** (Aliyun) - Chinese
  - Compute, Analystics, Network, Storage, IOT, Communication, Application Development, Data Migration
- **Amazon Web Services**

    - Compute, DevOps, IoT, Machine Learning, Networking, Robotics, Data Analytics, Content Delivery, Compute
- **Google Cloud**
    - Infrastructure, Platform, Serverless Computing, Communication, Collaboration, Productivity, Storage
- **IBM Cloud**
    - **Prominent Offerings**
        - Bare Metal Servers, VMWare Hosting, Virtual Private Cloud, Blockchain, AI (Watson)
    - **Products, Services**
        - Data & Analytics, Compute, Network, Storage, Management
- **Microsoft Azure**
    - Application and Services (Building, Testing, Deploying Managing)
- **Oracle Cloud**
    - Wide Ranging Applications - ERP, Cloud Database Services, Sales, Marketing, HCM, SCM
- **Salesforce** (Sales Cloud, Service Cloud, Marketing Cloud)
    - Analytics in real time, Customer Success and Support, Social Platforms
- **SAP**
    - Enterprise Software & Applications (ERP, CRM, HR, Finance)

Business Case for Cloud Computing

Cloud Adoption

- Businesses require speed, agility, innovation, decision making capacities

Cloud Adoption Case Studies

- Cloud improves customer service, creates faster time to market, efficient operations

---

Emerging Technologies Accelerated by Cloud

- **Internet of Things in the Cloud**
    - Devices - IoT Platforms - Cloud - Backend Analytics Platform
- **Artificial intelligence**
    - AI - IoT - Cloud
- **Blockchain & Analytics**
    - Blockchain - Cloud - Artificial Intelligence

    - Blockchain - a secure, immutable network that allows members to view only the transactions that are relevant to them
    - Artificial Intelligence - powers the analytics and decision making from the data collected
    - Cloud - provides globally distributed, scalable, and cost-efficient computing resources
- **Analytics**
    - Track social media trends, interpret data for machine learning, utilize data for predictive solutions (example: maintenance for infrastructure)
- **CLOUD SERVICES**
    - IoT (Devices)--Message Hub--Cloud Functions--Cloud Object Storage
    - **<u>ARTIFICIAL INTELLIGENCE, Big Data PLATFORMS</u> - Q&A data analytics utilizing machine learning**
        - **<u>General</u>** machine learning:
            - Apple CoreML, Amazon Machine Learning IBM Watson,
        - **<u>Specific</u>** machine learning:
            - Google Deepmind (machine learningneuroscience)
            - Baidu (Google's Chinese equivalent company) - created Baidu Minwa (mirrors ibm's watson)

Part 2

CLOUD COMPUTING MODELS
Overview:
Cloud Computing - Data Storage - Coding Languages - Programming - Software Development - Machine Learning (Analytics, Algorithms) - Artificial Intelligence

Architecture, Infrastructure and Security for each element

CLOUD COMPUTING MODELS

SERVICE MODELS

- **Cloud Service Market**
  - **IAAS** - **infrastructure as a service (SYSTEM ADMIN)**
    - **The entire cloud**
  - **PAAS** - **platform as a service (Dev, Jane - IBM)**
    - **The cloud software in tandem with the infrastructure located offsite**
  - **SAAS** - **software as a service (SOFTWARE,** anyone)
    - **Just the cloud software**

- **IAAS** - Cloud Provider Manages...(similar to leasing a car)
  - Data centers, Cooling, Power, Networking & security, Servers & Storage
- **PAAS** - Cloud Provider Manages...(similar to renting a car)
  - Data centers, Cooling, Power, Networking & security, Servers & Storage
  - Operating System, Development tools, Databases, Business Analytics
- **SAAS** - Cloud Provider Manages... (similar to a taxi)
  - Applications, Data

- **Infrastructure as a Service (IAAS)** - Compute, Networking, Storage Resources virtualized by a vendor for users
  - Provided on-demand, and pay as you go basis over the internet
  - Cloud provider hosts the infrastructure components in an on premises data cents, as well as the virtualization or hypervisor layer
  - **Virtual Machines (VM's)** - customized machines with customized layers
    - Layers: Hypervisor, Middleware, Applications, Data

        - Data: storage: backups and workloads track and monitor performance of services in addition to disaster recovery
    - **Physical Data Centers -** physical machines (in most IAAS Models, end users do not interact with the physical infrastructure, server farms, ect.)
    - **Compute -** hypervisors (connecting multiple Virtual Machines) are managed by providers for computing, memory, and storage (auto scaling, load balancing) for scalability and high performance
    - **Network -** access to resources on the cloud through virtualization or API
    - **Storage - object, file, block**
        - **Object - most common form of storage (highly distributed and resilient)**

- **<u>IAAS (Infrastructure as a Service)</u>** - uses in the current industry
    - Test & Development - (enable faster environments for application development)
    - Developers - are able to focus on business logic (instead of infrastructure management, since the infrastructure has been outsourced to cloud providers))
    - Business Continuity - secures access to applications and data during disasters/outage
    - Faster Deployment / Scaling - web applications deployed more quickly, scale infrastructure up or down as demand fluctuates
    - High Performance Computing - use cloud to solve complex problems involving millions of variables and calculations (financial modeling, climate, weather predictions)
    - Bid Data Analysis - locate patterns trends, associations for large data sets

- **<u>Platform as a Service (PAAS)</u>** - provides complete application-to-platform (develop, deploy, run, manage applications), all the virtualized resources from IAAS managed by a company Admin
    - PaaS Providers provide installation, configuration, and operation of application infrastructure, giving the user manager the application code (API)
    - PaaS Providers host and manage: Servers, Networks, Storage, Operating System, Application Runtimes, API's, Middleware, Database
    - Scalability, faster time to market, middleware allows less coding, greater agility innovation (access to more tools with less resources),
    - PaaS Providers: AWS Elastic Beanstalk, CloudFoundry, IMB Cloud Paks, Windows Azure, Redhat Openshift, Magento Commerce Cloud, Force.com, Apache Stratos
    - Risks of PaaS - information security threats, dependency on service providers infrastructure, customers lack of control over changes in strategy, service offerings, or tools

- **PAAS (Platform as a Service)** - current industries using PaaS for build, test, deploy, enhance, and scale applications rapidly and cost effectively
    - API development and management
    - Internet of Things (IoT)
    - Business analytics/intelligence
    - Business Process Management (BPM)
    - Master Data Management (MDM)

- **Software as a Service (SAAS)** - a cloud offering that provides access to a service provider's cloud-based software. Software on machine (charged as subscription model typically)
    - SaaS Providers maintain: Servers, Databases, Application Code, Security
    - SaaS Providers manage: Access, Security, Availability, Performance
    - SaaS supports: email, collaboration, customer relationship management, human resource management, financial management
    - Multitenant Architecture, Manage Privileges and monitor date, security compliance and maintenance, customize applications, subscription model, scalable resources
    - Benefits: greatly reduce tim from decision to value in business, increase workforce productivity and efficiency (users can access core business apps from anywhere, businesses and users can buy and deploy apps in minutes), spread out software costs over time

- **SaaS (Software as a Service)** - uses in the current Industries
    - Reduce on-premises IT infrastructure and capital expenditure
    - Avoid ongoing upgrades, maintenance, and patching
    - Run applications with minimal input
    - Manage websites, marketing, sales, and operations
    - Gain resilience and business continuity of the cloud provider
    - Concerns: jdata ownership and data safety, third-party maintains business-critical data, needs good internet connection

Part 2

DEPLOYMENT MODELS

- **Public Cloud** - users get access to Servers, Storage, Network, Security, Applications (delivered by provider over the internet)
    - Public Cloud - a virtualized multi-tenant architecture enabling tenants or users to share computing resources
    - Cloud Provider: owns, manages, provisions, and maintains the infrastructure
    - Users: use web consoles, and API's for the resources and services they need
    - Public Cloud Providers: Amazon Web Services, Windows Azure, IBM Cloud, Google Cloud, Alibaba Cloud
    - Benefits: access to on-demand resources (resources capable of fluctuating to the demand - economies of scale) at an affordable price and the resources are highly reliable (all physical components are maintained offsite with backup components)
    - Concerns: security, data sovereignty
    - Uses for Public Cloud: building and testing applications, reducing time to market for products and services, business with fluctuating capacity and resources, build secondary infrastructure for disaster recovery, data protection, and business continuity, greater accessibility, easy distribution, backing up data, outsource the management of platforms and applications

- **Private Cloud** - cloud infrastructure provisioned for exclusive use by a single organization comprising multiple consumers, such as the business units within the organization. Private cloud may be owned, managed, and operated by the organization, a third party, or some combination, existing on premises or off premises.
    - Internal (on premises) or External platforms (off premises)
    - VPC Virtual Private Cloud - external cloud that offers a private, secure, computing environment in a shared public cloud
    - Private Clouds allow for Private Control of Access, Security, and Compliance (with the same benefits of public cloud: scalability, cost efficiency, self service)
    - Benefits: Controlled by Internal IT (minimized data leaks, reduced cost, customized scalability, greater agility saving time internally), Customized Security and Customized Access
    - Uses for Private Cloud: opportunity to modernize inhouse applications, integrate data and application services already within a company, build applications uniquely without compromising security or compliance, no restrictions on changing parameters of security due to full control of security

- **Hybrid Cloud** - Connects an organizations' on-premise private cloud and third-party public cloud into a single flexible infrastructure (streamlines resources between public and private clouds)
    - "Cloud Bursting" - a spike in demand for cloud services (hybrid cloud can leverage demand between both cloud resources, public and private to eliminate cloud bursting)
    - Interoperable (public and private infrastructure orchestrated to work together), Scalable, Portable (move data on and off premises)
    - Hybrid Mono Cloud (one cloud provider) vs Hybrid Multicloud (multiple providers for the public cloud, might not be able to move data across providers seamlessly), Composite Multicloud (move resources across multiple providers)
    - Benefits: Security and compliance, scalability and resilience, resource optimization, cost saving
    - Industries Using of Hybrid Cloud in other industries: Software as a service integration, data and AI integration, enhancing legacy applications, Virtual Machine ware migration (minimize on premises data footprint to be able to scale undetected)
    - Current applications are moving towards the hybrid cloud because the hybrid cloud is excellent for integrating (updating, maintaining, modernizing) legacy applications

---

PART 3

COMPONENTS OF CLOUD COMPUTING

Overview:

Cloud Computing - Data Storage - Coding Languages - Programming - Software Development - Machine Learning (Analytics, Algorithms) - Artificial Intelligence

Architecture, Infrastructure and Security for each element

COMPONENTS OF CLOUD COMPUTING

CLOUD INFRASTRUCTURE

Cloud Network	**On Premise**
Logical Instances	Physical Devices
vNics (virtual network interface controllers)	NICs (network interface controllers)
Networking functions as Service	Networking Equipment in physical racks

Size of Cloud Network defined by the number of IP Addresses

- **Regions, Zones, and Data Centers**
    - Region - IT environment distributed across geographic areas where a cloud providers infrastructure are clustered. Regions are isolated from each other to maintain continuity.
    - Zones (Availability Zones, AZ) - areas allocated within a region to maintain power, cooling and networking resources, zones are also isolated like regions. Zones are connected using very high bandwidth connectivity with other zones, data centers, and the internet
    - Data Centers - huge room or warehouse containing cloud infrastructure (pods, racks, servers, storage, networking equipment
        - Computing Resources - cloud providers offer virtual servers, bare metal servers, and serverless computing resources

- **Virtual Machines (VM)** - most servers in a cloud data center run hypervisors to create virtual servers (virtual machines), software based computers based on virtualization technologies

- **Bare Metal Servers** - physical machines (servers) that are not virtualized
- **Serverless** - an abstraction layer on top of virtual machines (virtual models of physical machines) - Essentially a software version of a server on top of a software based computer (virtual machine)
- **Storage** - Block Storage (traditional data storage), File Storage (traditional data storage), Object Storage (most common currently used form for its resiliency)

- **Networking** - routers, switches,
    - Software Defined Networking (for users) - allows for easier networking and enables provisioning, configuration, management
    - **Provisioning** - (connecting the servers to the internet publicly and privately respectively) servers on the cloud need to be setup for public and private interfaces
    - **Network Interfaces** - assigned IP addresses and subnets (configure which network resources can access resources), including Security Groups and Access Control Lists (ACL), Virtual Area Networks (VLAN), Virtual Private Clouds (VPC), and Virtual Private Networks (VPN)
    - Virtualized Services: firewalls, load balancers, gateways, and traffic analyzers can all be virtualized services in the cloud
    - Content Delivering Networks (CDN) - quicker access with cloud providers to content

- **Virtualization and Virtual Machines** - (Virtualization, Hypervisors, Benefits)
    - **Virtualization** - creating a software based version of something (compute, storage, networking, servers, applications)
        - Cost Savings - multiple VM's consolidated to one piece of infrastructure using a hypervisor
        - Agility, Speed - virtualizing a new virtual environment is quick and simple
        - Lowers Downtime - numerous backup plans in place using multiple VM's
    - **Hypervisor** - (makes virtualization possible) a hypervisor is a piece of software that runs above the physical server or host by pulling the resources from the physical server and allocating the resources to a virtual environment (2 types of hypervisors)
        - **Type 1 (Bare Metal Hypervisors)** - hypervisor that is installed directly on top of the physical server (bare metal hypervisors), most frequently used, most secure, lowest latency (examples: VMware, ESXi, Microsoft Hyper-v, open source KVM)
        - **Type 2 (Hosted Hypervisors)** - hypervisor that is installed with a layer of "Host OS" between the physical server and hypervisor (end user virtualization, less frequent, higher latency) (examples: Oracle, VirtualBox, VMware Workstation)
    - **Virtual Machine VM** (software based machines, multiple can be run on a hypervisor once the hypervisor is installed) run on an operating system

- **<u>Virtual Machines (VM's) (types)</u>**
    - Virtual Servers, Virtual Instances, Instances
    - **Creating Servers or Virtual Machines in the Cloud**
        - Specify Region and Zone server will be provisioned in and the operating system on it (examples: Unix, Windows OS, Linux)
        - Specify "Shared" Multi-tenant or "Dedicated" Single-tenant
        - Specify Billing (hourly or monthly)
    - **Share or Public Cloud VM's** - provider managed, multi-tenant deployments, provisioned on-demand with predefined sizes
        - Underlying physical server is virtualized and is shared across other tenants or users
        - Predefined Sizes: single virtual core and a small amount of RAM (random access memory) up to multiple virtual cores and large amounts of RAM
        - Examples of Predefined Public Cloud VM's: "compute intensive," "memory intensive," "high performance I/O"
    - **Transient or Spot Virtual Machines** (temporary VM's that use vacant space on servers) - excellent for non-production, testing and developing applications, running stateless workloads, testing scalability, running big data, and HPC workloads at a low cost
    - **Reserved Virtual Server Instances** - reserve capacity and guarantee resources for future deployments (example 1 - 3 years of reserved server cloud space)
    - **Dedicated Hosts: Single Tenant Isolation with Dedicated Host** - host server is reserved entirely for a sole proprietor of VM's for exclusive use and full capacity of the hardware (the physical server host)
        - Specify the data center and pod and specify virtual machines to that host, allows for maximum control over workload placement

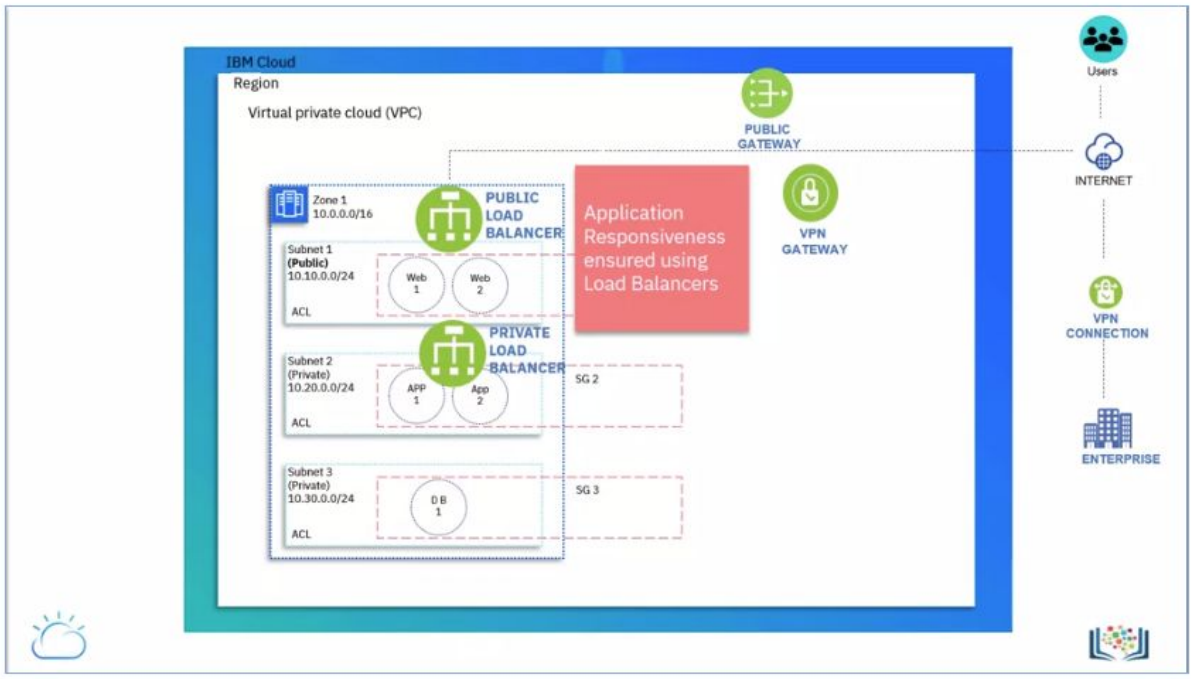
- **<u>Bare Metal Servers</u>** - **<u>(the REAL servers for REAL HIGHEST PERFORMANCE COMPUTING)</u>** a single-tenant, dedicated physical server (dedicated to a single customer). The cloud provider takes the physical server and plugs it into a rack in a data center for a single customer
    - Uses: High Performance Computing (HPC), Big Data Analytics, GPU-intensive solutions (graphics processing unit), ERP (enterprise resource planning), CRM (customer relationship management), AI Deep Learning, Virtualization
    - Many Cloud Providers do not offer Bare Metal Servers, they are intended for:
        - Fully customizable, demanding environments
        - Dedicated or long-term usage
        - High Performance Computing
        - Highly secure, isolated environments
    - The cloud provider manages the server up to the OS (operating system), they will reboot the server if anything physical maintenance is necessary.
    - Preconfigured (to meet workload packages) or Custom-configured

        - Processors, RAM (random access memory), Hard Drives, Specialized Components, The Operating System
    - **GPU (Graphics Processing Unit** - A programmable processor specialized for rendering all images on the computer's screen.**) GPU's can be added to bare metal servers for:**
        - Accelerating scientific computation
        - Data Analytics
        - Rendering professional grade virtualization graphics
    - Longer provisioning time (providing a server to a customer takes longer because they are PHYSICAL bare metal servers) (20-40 minutes for preconfigured, 3-4 hours for custom builds), bare metal servers are also more expensive than same size virtual machines
    - No hypervisor required: allowing clients full access

- **Secure Networking in the Cloud**
    - Digital Data and Cloud Environments require increased cybersecurity

      Virtual Private Cloud Security Infrastructure
    - Virtual Private Cloud (VPC) - security of private cloud, scalability of public cloud, virtual machines (vm's) and other resources are deployed into subnets
        - **Subnets** - allow virtual multi-tier services the same concept used in on premise environments, **subnets are where security is implemented in the cloud using Access Control Lists** (creates firewalls within the multi-tier cloud). Within subnets, security groups are created, creating **Virtual Server Infrastructure (VSI)**.
    - Cloud Network - > Subnets -> Virtual Server Infrastructure (VSI) -> Application (building, developing, testing, ect.)
    - **Subnet -> VSI -> Storage**
        - Example: 3 Tier Application:
            - Web Access VSI's, Application Tier VSI's, Backend Database VSI's

Secure Networking in Cloud

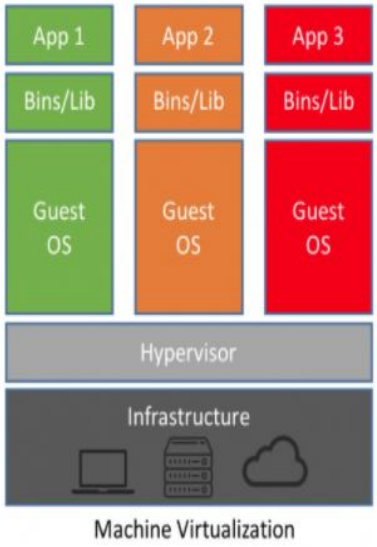


- 3 Security Groups (in this example, one for each)
- Public Gateway (for internet access) and VPN Gateway added for security
- Workload (Load Balancers) balancers are added to ensure the availability of bandwidth for the different applications within each group of VSI's (virtual server infrastructures)
- **Hybrid Cloud Security Infrastructure** - use dedicated high-speed, secure, direct private connectivity instead of public connectivity

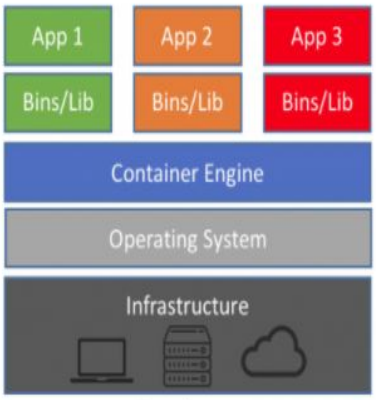
- **Building a Cloud Network**: entails creating a set of logical constructs that deliver networking functionality akin to data center networks for securing environments and ensuring high performing business applications.

- **Containers** - executable unit of software in which application code is packaged along with its libraries and dependencies in common ways so that it can be run anywhere. (similar to virtual machines)
    - **Containerization -** (Container applications developed significantly in 2008 Linux kernel introduced C Groups and control groups)
    - **Example** node.js application
        - Virtual Machine - hardware, host operating system, hypervisor
            - Subnet - Node.JS Application, Guest OS (operating system) (in this case linux)
            - Duplicate Subnets (2 for example)
        - Containers
            - Manifest Channel -> (ACI) Container Image -> Container (including all libraries, ect.)
            - Host Operating system, Runtime Engine (instead of Hypervisor)
            - Subnet/Containers - 3 copies of the application scaled out for example

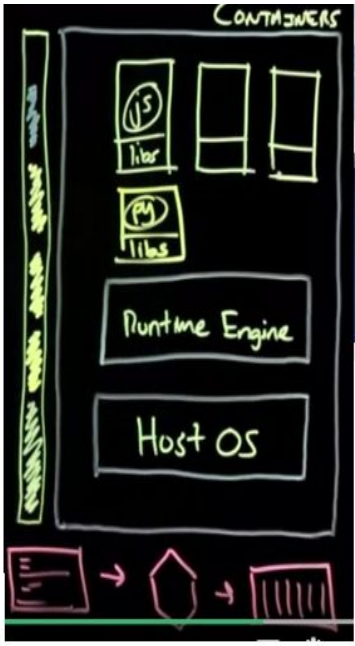
Virtual machines require an extra operating system, containers eliminate the need for guest OS



Machine Virtualization



Containers



But this approach has had its drawbacks. Each VM includes a separate operating system image, which adds overhead in memory and storage footprint. As it turns out, this issue adds complexity to all stages of a software development lifecycle—from development and test to production and disaster recovery. This approach also severely limits the portability of applications between public clouds, private clouds, and traditional data centers.

Part 3

CLOUD STORAGE AND CONTENT DELIVERY NETWORKS

- **Basics of Storage on Cloud**
  - **IOPS** - Input/Output Operations Per Second (speed of the storage, how quickly data can be read from or written to the storage)
  - **Persistence** - what happens to the storage once the compute node is attached to is terminated (if storage is set to “persist” it will not be deleted)
  - **Snapshot** - a point in time image of the storage (fast to create, metadata, don’t actually write data, only record changes to the data, cannot recover individual files)
  - **Cloud Storage** - save data and files in the cloud
    - 4 types of storage: Direct Attached, File Storage, Block Storage, Object Storage
    - Cloud Providers: Host, Secure, Manage, Maintain the storage and infrastructure
    - Storage services scale to size and amount of storage, also, the faster the read/write speed the higher the per gigabyte cost

        - Compute Node - certain storage must be attached to a compute node (other types of non-cloud storage can be accessed through the public internet or a dedicated private network connection)
    - **Direct Attached (local storage)** - storage presented directly to a cloud based server and is directly attached within the same host server chassis or rack, fast, used for storing a server's OS (operating system)
        - Ephemeral (temporary) - only lasts as long as the compute resource it is attache to
        - It cannot be shared with other nodes
        - Not as resilient to failure as other types of storage
    - **File Storage (network file system storage)** - storage is connected to compute nodes, servers, the connected to the cloud (connected to compute nodes over an NFS network)
        - Desktop Users are familiar with File Storage due to its hierarchical folder structure (Most Common Place Storage Type)
        - Slower than direct attached or block storage although lower cost and is attached to multiple servers
    - **Block Storage** - storage presented to compute nodes with high speed connections (used for databases where disc speed is important). Stored in volumes, then mounted onto a compute node.
    - **Object Storage** - not attached to a compute node, accessed via an API (least expensive
        - Infinite is size to the end user, unlimited storage space, although it has the slowest read/write speed (IOPs)
        - Great for all types of data

CLOUD STORAGE AND CONTENT DELIVERY NETWORKS

- **File Storage**
    - Attached to a compute node to store data
    - Less expensive, more resilient to failure, less disk management and maintenance for user, large amounts of storage can be presented as a disc to a server
    - Physical Disks -> Storage Appliances -> Compute Node
    - File storage is mounted from remote storage appliances (storage appliances are resilient to failure and offer encryption in transit and encryption at rest - appliances managed by service provider)
    - File storage is mounted on compute nodes via ethernet networks, network dedicated for storage, network attached storage, or network file storage (these networks vary in network traffic speed and this type of storage is for workloads where consistent speed is not required)
    - Multiple Compute Nodes - file storage can be mounted onto more than one compute node (excellent for common workloads: departmental file sharing, 'landing zone' for incoming files, repository of files) - ideal if speed variance is not an issue, low cost database storage
    - **IOPS** - input/output operations per second (higher iops speed, faster read/write)

- Check application requirements for IOPS speed

- **Block Storage**
  - Block storage brakes files into chucks (or blocks) of data, stores each block separately under a unique address, must be attached to a compute node before it can be utilized
  - Mounted from remote storage appliances, extremely resilient to failure, data is more secure (encryption in transit and at rest)
  - Mounted as a volume to compute nodes using a dedicated network of optical fibres, signals move at the speed of light (more expensive to build), higher price-point due to faster speed although higher building cost, perfect for workloads that need low-latency
  - Mounted onto only one compute node at a time - consistent high speed, excellent for databases and mail storage, not suitable for shared storage between compute nodes (multi-compute node sharing requires disc space)
  - IOPS (must be considered) - specify IOPS characteristics, adjust IOPS as need, depending on requirements and usage behavior

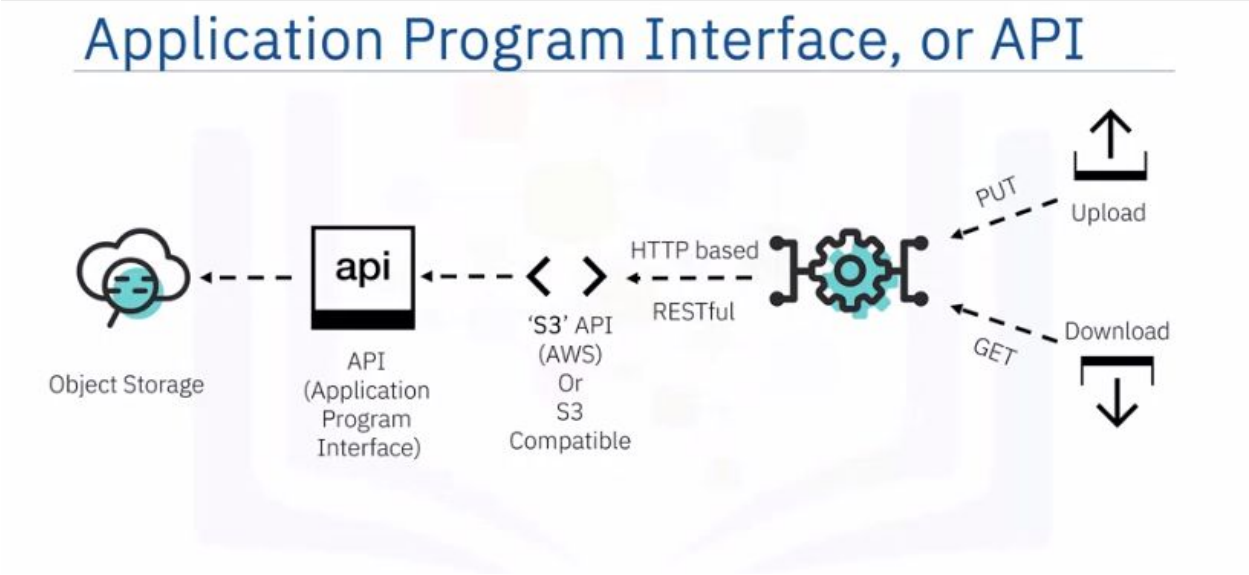
- **File Storage** - Ethernet network (attached to network) (stores at varying speeds) - good for file sharing (multiple compute node)
- **Block Storage** - High speed fibre network at high speeds - good for applications (requiring high speeds)

- **Object Storage Overview**
  - Object Storage - is not connected to a compute node, instead, provision an object storage service instance (virtual machine) and use an API (Application Program Interface) to upload, download, and manage data. Object storage can be used with any API and an underlying compute node is not necessary
  - Less expensive than other cloud storage options (per Gigabyte cents per month)
  - Infinite - only purchase amount needed (great for storing large amounts of unstructured data)
  - Object Storage uses “buckets” (like folder, objects stored in a structurally flat way, no buckets can be place in other buckets) unique metadata for each bucket id used to locate, access object, or when data was stored and accessed
  - **“Bucket”** - data storage format with no need to provide or define any sizing information (can hold few bytes to multiple petabytes), add data slowly or quickly or shrink it back down again, service provider manages resilience and making sure the bucket space is highly available (service providers offer different types of buckets)
    - Bucket Options: resilience (example: data is only stored in one data center, good if data needs to reside in one geographical location or if high

availability is less of an issue) (another example: data can be stored multiple times or in multiple locations, regions, and different data centers)
    - Object Storage Data - any data which is static and where fast read and write speeds are not necessary (text files, audio files, video files, IoT Data, VM images, backup files, data archive)
    - **STATIC** - CANNOT CHANGE - not suitable for operating systems, databases, or ANY CHANGING content

- **Object Storage - Data Tiers (or Classes) and API's**
    - **Object Storage Buckets** - associated with Tiers (Classes) based on how frequently they are accessed.
    - **Standard Tier** - stores objects that are frequently accessed, has the highest price per gigabyte cost
    - **Vault/Archive Tier** - data accessed only once or twice a month (or less), offered at a lower cost
    - **Cold Vault** - data accessed only once or twice a year (costs just fraction of a cent per gigabyte)

    - Automatic archiving rules for data - if data is not accessed, automatically moved to a cheaper storage tier determined by meta data
    - Object Storage does not have IOPS options
    - Slower in comparison to file and block storage
    - Costs of Object Storage: priced per gigabyte, costs related to retrieval of the data, higher access costs for cold vault tiers



    - Effective solution for backup and disaster recovery, replacement for offsite backups, many backup solutions come with built-in options for Object Storage on Cloud, more efficient than tape backups for geographic redun

- **Object Storage** - much slower than **File Storage** and **Block Storage**

- **Content Delivering Networks (CDN)**
    - Content Delivery Network - Distributed Server Network that delivers temporarily stored (cached) copies of website content based on geographic locations from distributed locations (reduces the distance between website visitor and website server)
    - Content Delivery Networks - service that accelerates internet content delivery and makes websites faster

ADDITIONAL (FROM WIKIPEDIA)
ABSTRACTION (DEFINITION)

- the creation of abstract concept-objects by mirroring common features or attributes of various non-abstract objects or systems of study[3] – the result of the process of abstraction.

Abstraction, in general, is a fundamental concept in computer science and software development.[4] The process of abstraction can also be referred to as **modeling** and is closely related to the concepts of *theory* and *design*.[5] Models can also be considered types of abstractions per their generalization of aspects of reality.

Abstraction in computer science is closely related to abstraction in mathematics due to their common focus on building abstractions as objects,[2] but is also related to other notions of abstraction used in other fields such as art.[3]

Abstractions may also refer to real-world objects and systems, rules of computational systems or rules of programming languages that carry or utilize features of abstraction itself, such as:

- the usage of data types to perform *data abstraction* to separate usage from working representations of data structures within programs;[6]
- the concept of procedures, functions, or subroutines which represent a specific of implementing control flow in programs;
- the rules commonly named "abstraction" that generalize expressions using free and bound variables in the various versions of lambda calculus;[7][8]
- the usage of S-expressions as an abstraction of data structures and programs in the Lisp programming language;[9]
- the process of reorganizing common behavior from non-abstract classes into "abstract classes" using inheritance to abstract over sub-classes as seen in the object-oriented C++ and Java programming languages.

PART 4

EMERGENT TRENDS AND PRACTICES

Overview:

Cloud Computing - Data Storage - Coding Languages - Programming - Software Development - Machine Learning (Analytics, Algorithms) - Artificial Intelligence

Architecture, Infrastructure and Security for each element

EMERGENT TRENDS AND PRACTICES

HYBRID MULTI-CLOUD, MICROSERVICES, AND SERVERLESS

- **Hybrid Multi-cloud**

- **Hybrid Cloud** - a computing environment that connects an organization's on-premise private cloud and third party public cloud into a single infrastructure for running the organization's applications
- **Multi-cloud** (mix of cloud models from different service providers - public, private, and managed - across infrastructure, platform, or software services)
- **Cloud Scaling** - adaptations of the cloud to meet an organization's business changes
    - Utilizing the cloud to decrease latency and meet increases and decreases in business models
- **Composite Cloud** - moving around aspects of an organizations business applications to different cloud models to attain unique qualities such as access to cloud scaling (example, moving business applications to a different region to in created cloud speeds to adjust for business increases)
    - Taking advantage of the cloud to accelerate business

- **<u>Microservices</u>**
    - **Microservices Architecture** - **applications with many components**. (a single application is composed of many loosely coupled and independently deployable smaller components or services (running on their own stack in their own containers - they communicate with API's, Event Streaming, and Message Brokers)
        - **Components** can be updated by multiple developers working independently
        - Teams can use different stacks and runtime environments
        - Independent scaling
    - **Cloud Development Platforms**
        - Provide an ecosystem of already existing code that can be easily integrated into applications
        - Allows Developers to use code already created (instead of building one huge application on one team), no longer creating applications from scratch
    - **MICROSERVICES** - developers break into small independent teams where they write smaller amounts of code called microservices.
        - **Microservices** break down large applications into their core functions (example: "search" "recommendations" "customer ratings" "product catalog")
        - **Container** - distribution method for each microservice, delivers the code where it needs to go
            - Containers are plug and play - so if one isn't working developers can take it out and put in a different one without disrupting the rest of the application

- **Serverless Computing**
  - **There are still servers -- the management is removed -- management of the underlying physical or virtual servers is removed from their users.**
  - **Serverless** - an approach to computing that offloads responsibility for common infrastructure management tasks such as: scaling, scheduling, patching, and provisioning application stacks to cloud providers allowing developers to focus on the code specific to their applications or process
    - Allocates resources as needed for applications
  - Key Attributes of Serverless Compute Models
    - No provisioning of servers & runtimes
      - (No provisioning of servers, installation of application stacks, and software. No provisioning operation of infrastructure by the developer.)
    - Runs code on-demand, scaling as needed
    - Pay only when invoked and used (i.e. not when underlying computer resources are idle) - (unlike virtual servers on the cloud, where end users pay for vm's as long as they are running)
    - Abstracts the infrastructure away from developers
    - Code executed as individual functions inside an individual container
    - No prior execution context is required to serve a request
  - **Serverless Computing Services**
    - **IBM Cloud Functions**
    - **Amazon Web Services Lambda**
    - **Microsoft Azure Functions**
  - Determining Fit with Serverless
    - Evaluate application characteristics
    - Ensure that the application is aligned to serverless architecture patterns
    - Applications that qualify for a serverless architecture:
      - Short-running stateless functions
      - Seasonal workloads
      - Production volumetric data
      - Event-based processing
      - Stateless microservices
  - Use Cases - Serverless architectures are well-suited for use cases around: data and even processing, IoT, Microservices, and Mobile backends
  - Use Cases - well-suited to working with: text, audio, image, video
    - **Tasks:** data enrichment, transformation, validation and cleansing, pdf processing, audio normalization, thumbnail generation, video transcoding, data search and processing, genome processing
    - **Data Streams**: business, IoT sensor data, Log data, financial market data
  - Challenges
    - Workloads characterized by long-running processes are better for traditional servers (simpler, more cost effective)

        - Low-latency applications requiring no delay need actual servers (such as financial applications

---

Part 4

CLOUD NATIVE APPLICATIONS, DEVOPS, AND APPLICATION MODERNIZATION

THE FUTURE OF APPLICATIONS IN INFORMATION TECHNOLOGY!!!!!!!!!
ALL APPLICATIONS ARE MOVING TO THE CLOUD!!!!
IN THE FORM OF MICROSERVICES AND CONTAINERS

- **Cloud Native Applications**
    - **Cloud Native Application** - an application developed to work only in the cloud environment (refactored and reconfigured with cloud native principles), microservices working together as a whole to comprise an application (independent, scalable, in containers)
    - **Development Principles** - whether creating a new cloud native application or modernizing an existing application
        - Microservices Architecture - breaking applications down into single function microservices
        - Rely on Containers - maximum flexibility, scalability, portability
        - Adopt Agile Methods - speed the creation and improvement process through quick iterative updates based on user feedback

Cloud Solution Stack (Layers) :

- **Cloud Native Applications - > microservices - > living on the cloud**
    - **Application Code** - **cloud native applications layer**
    - **Application Runtime** - middleware
    - **Application, Data Services** - backing services, integrate code with existing services on cloud and premise
    - **Scheduling, Orchestration** - control planes (kubernetes****)
    - **Cloud Infrastructure** - Private, Public, and Enterprise

- **Benefits** - Enterprise and Engineering at Scale: innovation, business agility, commoditizations of cloud application solution stack - core services have a lower center of gravity (infrastructure is less important, innovation is more important)

- **Use Cases** - The “Commoditization” (turning everything into a commodity, or product) “Everything” - everything that lives in the cloud should have a cloud native application **DESIGN** and **APPROACH,** application code needs to be instrumented with:
  - **Application Code Layer:** Standardized Logging, Standardized Events, Standardized Catalogue: for multiple microservices and cloud services to use
  - **Scheduling, Orchestration Layer:** Distributed Tracing (microservices - a lot of moving parts need to be tracked in scheduling and orchestration) - load balancing, routing,

Kubernetes is an open-source container-orchestration system for automating computer application deployment, scaling, and management. It was originally designed by Google and is now maintained by the Cloud Native Computing Foundation

- **DevOps on the Cloud**
  - Development Teams - design, develop deliver, and run software
  - Operations Teams - (identify and resolve problems by…) monitoring, predicting failure, managing environment, fixing issues
  - **DevOps** - a collaborative approach that allows multiple stakeholders to collaborate: business owners, development, operations, quality assurance
    - Developers can produce software in short iterations
    - A continuous delivery schedule of new features and bug fixes in rapid cycles
    - Businesses can seize market opportunities
    - Accelerated customer feedback into products
  - **The DevOps Approach**
    - A DevOps approach applies agile and lean thinking principles to all stakeholders in an organization who develop, operate, or benefit from the business’s software systems, including customers, suppliers, partners. By extending lean principles across the software supply chain, DevOps capabilities improve productivity through accelerated customer feedback cycles, unified measurements and collaboration across an enterprise, and reduced overhead, duplication, and rework.
  - **DevOps Process**
    - Continuous Delivery
    - Continuous Integration
    - Continuous Deployment
    - Continuous Monitoring
    - Delivery Pipeline
      - Ideation > Coding > Building > Deploying > Managing > Continuous Improvement >Ideation (continuous cycle)
  - **DevOps and Cloud**

        - Near limitless compute power and available data and application services, cloud computing platforms come with their own risks and challenges: Tools, Practices, Processes.
    - **DevOps Cloud Solutions**
        - Automated provisioning and installation
            - Build middleware, install application code, provision servers, fully automate installation process in a documented way
        - Continuous integration and deployment pipelines
            - Fully automated development pipeline
        - Define how people work together and collaborate
            - Build, deploy, manage applications in a cloud native approach (productivity and quality)
        - Test in low-cost, production-like environments
        - Recover from disasters by rebuilding systems quickly, reliably

- **Application Modernization**

    - **Enterprise Applications** - Monolithic Legacy Applications
        - Siloed Systems, Difficult to Update, Expensive to Maintain
    - **Modernized Applications**
        - Accelerate digital transformation, leverage new tech & services, respond faster to change

    - **Application Modernization**

Enterprise applications		**Transformation**: to Modernized Applications on Cloud		
Architecture	Monoliths	**Architecture**	Service Orientated	Microservices
Infrastructure	Physical Servers	**Infrastructure**	Virtual Machine	Cloud
Delivery	Waterfall long phases	**Delivery**	Agile Development	DevOps

---

PART 5

CLOUD SECURITY AND MONITORING, CAREER OPPORTUNITIES AND JOBS

Overview:

Cloud Computing - Data Storage - Coding Languages - Programming - Software Development - Machine Learning (Analytics, Algorithms) - Artificial Intelligence

Architecture, Infrastructure and Security for each element

CLOUD SECURITY AND MONITORING, CAREER OPPORTUNITIES AND JOBS

CLOUD SECURITY AND MONITORING

- **Cloud Security**
    - **Managing the risk and compliance of workloads and the data brought to the cloud.**

- **Shared responsibility** creating a **data security architecture** to **protect data** and **manage access**
  - Data - at rest, in motion, and in use (in memory, of application)
  - API Keys - more secure than only using encryption

- **PaaS** - user manages securing applications, workload, and the data, while the cloud provider manages security of the platform (compliance, security, - network, containers, runtime, and isolation/space within platform)
- **IaaS** - cloud provider manages the hypervisor and the servers, user manages the virtual servers, virtual machines, or everything if using bare metal servers
- **SaaS** - cloud provider manages all the applications and app security

- **Data Security Architecture**
  - Data Type - public, private (sensitive, confidential)
  - Application Type - building, migrating, or modernizing
  - Data encryption (for data at rest in a database, object store, or block storage)
  - API Keys - allow user to own the data and have complete control (by bringing own keys, and keeping own keys)
  - Container security - scan container images for vulnerabilities

- **Data Security Architecture**
  - **Manage Access**: Gain Insights: Posture, Compliance, Threats
    - Deployment Environment provides: Security Events, Audit logs, Flow logs (from network or system)
  - **Protect Data**: mediate on actionable intelligence
    - Replace vulnerable containers, block suspicious access (suspicious IP address)

- **Sec (Security) DevOps** (development and operations):
  - **Design** -> **Build** -> **Manage Security** -----> OLD WAY
  - **Secure Design** -> **Secure Build** -> **Re-manage Security** (new way)
  - Security FIRST (before Design and Build)
    - Old way was to Design, Build, Manage, AND THEN ADD Security: Application Teams: Design and Architecture (for building code), Enterprise Security Teams: secure and manage
    - **Secure Design** - plan for data type, level of classification, application types, container based or not
    - **Secure Build** - embedded security, security aware applications (data encryption before and after storage)
    - **Remediate Security** - rearchitect security design and build

FIRST LINE OF DEFENSE - FOR CLOUD COMPUTING

IDENTITY AND ACCESS MANAGEMENT

- **Identity and Access Management (Access Control)**
  - Cloud Security Concerns
    - Data Loss and Leakage
    - Unauthorized Access - misuse of employee credentials, improper access controls
    - Insecure Interface and APIs
  - First Line of Defense, authenticates and authorizes users, provides user-specific access
    - User types: (organizational, internet and social-based, third-party business partner organizations, vendors)
      - **Administrative Users** - (administrators, operators, managers) roles that typically create, update, and delete application and instances, and also need insight into their team members' activities
      - **Developer Users** - (application developers, platform developers, application publishers) read sensitive information, create applications, update applications, delete applications
      - **Application Users** - users of the cloud-hosted applications
  - **Unauthorized Access**
    - Attacker with administrative access: steal data from databases, deploy malicious applications, deface or destroy existing applications
    -
  - **Authentication & User Identity**
    - API Key (Application Programming Interface key) - a unique identifier used to authenticate a user, developer, or usually authenticate a project
    - Cloud directory, Social login, Enterprise identity provider, cloud identity provider
  - **Multifactor Authentication** - used to combat identity theft by adding an additional level of authentication for application users
    - Single use password, pins, certificates, tokens, or risk based (changes in the user's location, past activities, and preferences)
  - **Cloud Directory Services** - used to securely manage user profiles and their associated credentials and password policy inside a cloud environment (applications hosted on the cloud do not need to use their own user repository)
  - **Reporting**: helps provide a user-centric view of access to resources or a resource-centric view of access by users (keeps a record and shows which user accessed which resources)
    - Which users can access which resources
    - Changes in user access rights
    - Access methods used by each user
  - **Audit and Compliance** - critical service within identity and access management framework, both for cloud provider, and cloud consumer

        - Auditors - validate implemented controls against: security policy, industry compliance, risk policies, and report deviations
    - **User and Service Access Management** - enables cloud application-service owners to provision and de-provision
        - Streamlines acces control based on: role, organization, and access policy
        - App Owner -> provision, de-provision ->
            - Customer Profile
            - Partner Profile
            - Vendor Profile
    - **Mitigating Risks** - in order to mitigate the risks of these accounts being hacked into, you require maximum control over the whole life cycle of these users. Some of the controls that can help secure these sensitive accounts include:
        - Provisioning users by specifying roles on resources for each user
        - Password policies that control the usage of special characters, minium password lengths, and other similar settings
        - Multifactor authentication like time-based one-time passwords
        - Immediate de-provisioning of access when users leave or change roles
    - **Access Groups** - offered by most cloud providers
        - Create Access Groups, add users to access groups, manage access for existing users
        - A group of users and service IDs created so that the same access can be assigned to all entities within the group with one or more access policies
    - **Access Policies** - define how users, service IDs, and access groups in the account are given permission to access account resources. Policies include:
        - Subject - which can be users, service IDs, or access groups
        - Target - which is the resource, or provisioned service offering, to which you want to provide access
        - Role - the resource to which the access is being granted
    - **Access Group Benefits** - streamline access assignment process, as opposed to assigning individual user access. Also, reduces the number of policies in an account

SECOND LINE OF DEFENSE - FOR CLOUD COMPUTING

CLOUD ENCRYPTION

- **Cloud Encryption**
    - Encryption: plays a key role on Cloud, often referred to as the last line of defense in a layered security model
        - Encrypts data, provides robust data access control, key management, certificate management
        - Encryption: (definition) scrambling data in a way that makes it illegible, ensures only authorized users have access to sensitive data. When data is accessed without authorization, data is unreadable and meaningless.

        - Encryption Algorithm - defines the rules by which data will be transformed.
        - Decryption Key - define how encrypted data will be transformed back to legible data.
- **Cloud Encryption Services** - can be limited to encryption of data that is identified as sensitive, or end-to-end encryption of all data uploaded to the cloud (data is encrypted upon receipt, and decryption keys are provided to customers to decrypt data when needed). Keys need to be managed securely, losing keys will result in no being able to read data.
- **Data Protection States**
    - **Encryption at Rest** - protects stored data,
        - multiple encryption options: block and file storage, built-in for object storage, database encryption
    - **Encryption in Transit** - protects data while transmitting
        - Includes encrypting before transmission
        - Authenticates endpoints
        - Decrypts data on arrival
        - **Protocol for Encryption in Transit:**
            - **Secure Sockets Layer (SSL)** - used when accessing website securely, also for data moving between servers and services within the cloud
            - **Transport Layer Security (TLS)** - used when accessing website securely, also for data moving between servers and services within the cloud
    - **Encryption in Use** - protects data in use in memory (for computations)
        - Allos computations to be performed on encrypted text without decryption
- **Client or Server-side (of Cloud Encryption)**
    - **Server-side** - occurs after cloud server receives client data but before the data is written to disk and stored
        - Create and manage your own encryption keys
        - **Customer Managed Encryption Keys** - Generate and manage keys on cloud (key management services offered by the cloud storage provider)
    - **Client-side** - occurs before data is sent to cloud, users can utilize encryption keys and algorithms that are not visible to the cloud provider
        - Cloud providers cannot decrypt hosted data
- **Cloud Encryption Strategy:** A majority of enterprises today operate in multicloud environments, there is a need to implement a singular data protection strategy across an enterprise's on-premise, hybrid, and multi-cloud deployments.
- **Multi-Cloud Data Encryption** - features data access management, integrated key management, and sophisticated encryption ----> all of which combine to deliver scalability and flexibility to protect the most sensitive workloads across the enterprise regardless of where the data resides

        - **Multi-cloud data encryption console**
            - define and manage access policies
            - Create, rotate, and manage keys
            - Aggregate access logs
    - **Key Management** - encryption does not eliminate security risk. Key management separates the security risk from the data itself. These keys need to be managed and protected against threats.
    - **Key Management Services** - enable customers to encrypt sensitive data at rest, easily create and manage the entire lifecycle of cryptographic keys, protects data from cloud service providers as well as from other users.
        - Storing encryption keys separately from the encrypted data
        - Taking key backups offsite and auditing them regularly
        - Refreshing the keys periodically
        - Implementing multi-factor authentication for both the master and recovery keys

- **<u>Cloud Monitory Basics and Benefits</u>**
    - **Cloud Monitoring** - needs to be a priority for organizations looking to leverage the benefits of cloud technologies.-
    - **Cloud Monitoring Solutions** - monitoring performance across an entire stack of applications and services can be time-consuming and draining on internal resources.
    - **Cloud Monitoring Assessment** - assess data, application, and infrastructure behaviors for: performance, compliance, resource allocation, network availability, and security risk. More than automated tools, includes: Strategies, Practices, Processes all of which are used for: analyzing, tracking, and managing services and applications
        - Cloud Monitoring serves to provide actionable insights that can help improve availability and user experience
        - Accelerate the diagnosis and resolution of performance incidents
        - Control the cost of monitoring infrastructure
        - Mitigate the impact of abnormal situations with proactive notifications
        - Get critical Kubernetes and container insights for dynamic microservice monitoring
        - Troubleshoot applications and infrastructure
    - **Cloud Monitoring Solutions** - gives organizations visibility and control over entire cloud based infrastructure
        - Data in real-time with round the clock monitoring of Virtual Machines, Services, databases, and applications
        - Multilayer visibility into application, user, and file access behavior across all applications

        - Advanced reporting and auditing capabilities for ensuring regulatory standards
        - Large-scale performance monitoring integrations across multicloud and hybrid cloud
    - **Cloud Monitoring Categories** (infrastructure, database, applications performance monitoring)
        - **Infrastructure**: help identify minor and large-scale failures so that developers can take corrective action
        - **Database**: help track processes, queries, and availability of services, to ensure accuracy and reliability
        - **Application Performance** - (measures application availability and performance, providing tools needed to troubleshoot issues in an applications environment) help improve user experience, meet application and user SLAs, minimize downtime and lower operational costs
    - **Cloud Monitoring Best Practices**
        - 1) Leverage end-user experience monitoring solutions
        - 2) Move all aspects of infrastructure under one monitoring platform
        - 3) Use monitoring tools that help track usage and cost
        - 4) Increase cloud monitoring automation
        - 5) Stimulate outages and breach scenarios

Part 5

Career Opportunities and Job Roles in Cloud Computing

- **Career Opportunities and Job Roles in Cloud Computing**
    - Cloud Computing - a key part of an enterprise's digital transformation strategy
        - Companies are moving business process and applications to a mix of cloud infrastructures
        - Cloud Market - 182.4 Billion USD 2018 ---> to 331.2 Billion USD 2022
            - Growing at 3 times the market of IT Services
    - Cloud Specialization Areas:
        - Cloud Developers
        - Cloud Integration Specialists

        - Cloud Data Engineers
        - Cloud Security Engineers
        - Cloud DevOps Engineers
        - Cloud Solutions Architects
    - Cloud Software Engineer - software development lifecycle (writing, testing, maintaining, app front and back end, platforms and systems)
        - Experience with: IBM Cloud, Google Cloud Platform, Amazon Webservices, Microsoft Azure
        - Knowledge of: data structures, distributed systems, OS, Algorithms
        - Experience with Databases
        - Proficiency in web app languages (python, javascript, java, html, css)
    - Cloud Integration Specialist - integrate new cloud services, applications and infrastructure with existing systems and services
        - Assess implications and trade-offs between different solutions
        - Optimize integration
        - Optimize user experience
        - Optimize performance standards to meet Service Level Agreements
    - Cloud Data Engineer - designing, developing, deploying scalable data pipelines and data services
        - Integrate new data management technologies and software engineering tools into existing infrastructure
        - Recommend automated integration of disparate data sets
        - Collaborate with Data Scientists to create predictive models & PoCs
        - Promote best practices for consumption and understanding of data
        - Improve efficiency by introducing new engineering processes & tools
    - Cloud Security Engineer - protects organizations systems and application data (confidentiality, integrity, availability)
        - Determine security requirements
        - Plan, implement, and test security systems
        - Recommend innovative technologies to enhance security of cloud-based environments
        - In-depth knowledge of cloud platforms and services, software design patters and DevOps tools and methodologies
    - Cloud DevOps Engineer - create reliable and rapid release pipelines for software and updates
        - Create custom automation tools
        - Build and maintain configuration and deployment frameworks
        - Track and design bugs and automate the debugging process
        - Maintain and deploy web-based applications
        - Monitor security systems
        - Measure performance against expected business outcomes
    - Cloud Solutions Architect - translate business requirements into application architecture and design
        - In-depth understanding of cloud platforms and services

        - In-depth understanding of software design patters
        - Knowledge of DevOps tools and methodologies
        - Good understanding of networking
        - High level understanding of security concepts
        - Collaborate with Cloud Developers, Networking Specialists, Security Engineers, Integration Specialists and DevOps Engineers

    - Resources for Learning Cloud Computing: courses, videos, books, forums (cloud providers have learning portals: leaning paths, hands on learning labs, free trials)

NOTES FOR LECTURE SERIES:
Link in video description

Presentation on Cloud Computing for Business, Machine Learning and Artificial Intelligence

Cloud Computing - Data Storage - Coding Languages - Programming - Software Development - Machine Learning (Analytics, Algorithms) - Artificial Intelligence

Architecture, Infrastructure and Security for each element

Lecture By Daniel James Higgins
Presentation By Daniel James Films

Thank you.