**Cloud Computing**

NIST definition – model for enabling convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction

**5 essential characteristics**

1. On-demand self service
   1. Get access to cloud resources using a simple interface without need for human interaction
2. Broad Network Access
   1. Resources can be accessed via network through phones/tablets/laptops/etc.
3. Resource pooling
   1. Gives cloud providers economies of scale, which is passed on to customers
4. Rapid elasticity
   1. Access more resources when needed, scaled back when not needed
5. Measured service
   1. Only pay for what you use, reserve as you go

Cloud Computing as a Service

Cost-efficient, more agile companies

**3 Deployment models**

1. Public
   1. Leverage cloud services over open internet, usage shared by other companies
2. Private
   1. Infrastructure provisioned for exclusive use by a single organization, or owned/managed/operated by a service provider
3. Hybrid
   1. Mix of both public and private, working together seamlessly

**3 service models**

1. Infrastructure (IaaS – Infrastructure as a Service)
   1. Access to infrastructure and physical computing resources, without need to manage or operate them.
2. Platform (PaaS)
   1. Access to the platform tools, those needed to develop and deploy applications to users over the internet
3. Application (SaaS – software as a service)
   1. Software licensing and delivery model, centrally hosted software and licensed on a subscription basis. (On-Demand software)

**History**

CapEx (capital expense) moved to OpEx (operation expense)

**Key considerations for moving to cloud computing**

Agility, Flexibility, Competitiveness

Infrastructure and workloads – Pay as you go

SaaS and dev platforms

Speed and productivity

Risk exposure

Benefits of cloud adoption:

1. Flexibility – scale back and scale up as needed, customize applications, access cloud anywhere
   1. Security - Virtual private clouds, encryption, and API keys
2. Efficiency – apps to market quickly, accessible anywhere, hardware failures do not result in data loss due to backups
3. Strategic value
4. Challenges – Data security, governance issues, legal issues, lack of standardization, choosing the right deployment model, partnering with the right cloud service, concerns with business continuity

**Key Cloud Service Providers**

Alibaba Cloud - Aliyun

AWS Cloud – metered pay as you go

Google Cloud – GCP (serverless computing, GSuite), Developing and hosting software

IBM Cloud – Full stack cloud platform

Microsoft Azure

Oracle Cloud – SaaS, ERP, etc

Salesforce – focuses on business to customer services

SAP – Enterprise software, ERP, CRM, HR, Finance

**Business case for cloud computing**

Greater freedom to change course if failure is achieved

Case studies:

1. American Airlines
   1. Removed large legacy code applications in favor of microservices
   2. American Airlines adopting cloud technologies to deliver customer value rapidly across its enterprise
2. UBank
   1. PaaS – give more control to their developers, faster time to market
   2. UBank leveraging cloud platform services to give more control to their developers thereby removing barriers to innovation
3. Bitly
   1. Establish scalable, low latency software
   2. Bitly leveraging the scalability offered by cloud infrastructure for low-latency delivery to its geographically disbursed enterprise customers
4. ActivTrades
   1. Accelerate execution, streamline delivery of new functions, cut latency
   2. Migrated to IBM cloud VMWare solutions
   3. ActivTrades leveraging the infrastructure, storage, network, and security offerings on the cloud to accelerate execution and delivery of new functions in their online trading systems to their customers

**Internet of Things (IoT)**

Giant network of connecting things

Smart devices and sensors allow this happen

This data stored and processed on the cloud

Case Study:

IoT predicts poaching, allowing for the ability to stop it before it happens

**Artificial Intelligence**

Helps to digest the data gathered using IoT

AI helps to make decisions based on given data

IBM AI – Watson

**BlockChain and Analytics on the Cloud**

Blockchain – a secure, immutable network that allows members to view only the transactions that are relevant to them

3-way relationship

1. Blockchain provides trusted, decentralized source of truth
2. AI powers analytics and decision-making
3. Cloud provides globally distributed, scalable, and cost-efficient computing resources

Blockchain records data and variables that go into algorithm decisions

Tracking trends on Social media, analyze data to build machine learning models, data analytics and predictive maintenance solutions for city infrastructure

Case Study:  
Track lettuce/food chain to reduce waste when a recall happens. Don’t have to take “all” food off shelves because we can track where each batch came from

**Week 2 – Service Models**

IaaS, PaaS, SaaS

IaaS – persona is a system administrator or IT admin

SaaS – Persona can be anyone as a user.

PaaS – abstracts away certain things, Persona is a developer

Car example:

IaaS is like leasing the car, basically buying everything. Care about color/specs/etc.

PaaS is like renting a car, don’t care about color/specs, only paying for tolls/gas

SaaS is like getting a taxi/uber, don’t care about anything about the car. Only paying for the ride

**IaaS -** provides the fundamental compute, network, and storage resources for customers on-demand.

Form of cloud computing that delivers fundamental compute, network, and storage to consumers on-demand, over the internet, and on a pay-as-you-go basis.

Cloud provider hosts infrastructure components

Costumers can create or provision VMs wherever they want

-VMs are created for the customer, and they can then deploy their own middleware/applications/data/etc.

Cloud providers give customers ability to track and monitor the performance and usage of their cloud services and manage disaster recovery

Key components:

1. Physical data structures
   1. End users do not interact directly with physical structures
2. Compute – users provision their own resources with desired amounts of compute, memory, and storage
   1. Comes with supporting services like auto scaling, load balancing, both of which provide scalability and high performance
3. Network
   1. Access to network through virtualization or APIs
4. Storage
   1. Object (most common given that it’s highly distributed and resilient), File, block storage

Use cases:

1. Test and development environments, abstracting low level components
2. Business continuity and disaster recovery, reduce cost and make apps and data accessible as usual during a disaster scenario
3. Faster deployments and scaling
4. High performance computing – solve complex problems
5. Big data analysis – cloud provides computing and processing power

Concerns: lack of transparency, dependency on a third party

**PaaS -** provides customers the hardware, software, and infrastructure to develop, deploy, manage, and run applications created by them or acquired from a third-party.

Cloud computing model that provides customers a complete platform to develop, deploy, manage and run applications.

PaaS provider hosts and manages: servers, networks, storage, OS, application runtimes, API, middleware, databases, etc.

Takes responsibility for installation, configuration and operation.

User only responsible for app code and maintenance

Abstracts user from lower level details

Essential characteristics:

1. High level of abstraction provided to the users, eliminate complexity of deploying applications
2. Provides support services and APIs to simplify the job of devs, including security
3. Run-time environments – executes code according to application owner and cloud provider policies
4. Provide rapid deployment mechanisms
5. Middleware capabilities – app servers, DB management, mobile back-end, event processing

Use Cases:

1. API dev and management
2. IoT
3. Business analytics/intelligence
4. Business process management (BPM)
5. Master data management (MGM)

Advantages:

1. Scalability
2. Faster time to market
3. Reduce amount of code needed – greater agility and innovation

Available offerings:

1. AWS Elastic Beanstalk
2. CloudFoundry
3. IBM Cloud Paks
4. Azure
5. OpenShift
6. Magento
7. Force.com
8. Apache Stratos

Risk of PaaS

1. Info sec threats
2. Dependency on service providers infrastructure
3. Customers lack control over changes in strategy, service offerings, or tools

**SaaS -** provides access to users to a service provider’s cloud-based software. Users simply access the applications on Cloud while the Cloud provider maintains the infrastructure, platform, data, application code, security, availability, and performance of the application.

A cloud offering that provides access to a service provider’s cloud-based software

Maintains: servers, databases, code, and security

SaaS Supports:

1. Email and collaboration – Office 365
2. Customer Relationship management (CRM) – netsuite CRM, salesforce
3. Human Resources management – SAP
4. Financial Management

Key Characteristics:

1. Multitenant architecture
2. Manage privileges and monitor data
3. Security, compliance, maintenance
4. Customize application UI
5. Subscription model
6. Scalable resources

Key benefits

1. Greatly reduce the time from decision to value from months to days
2. Increase workforce productivity and efficiency
   1. Users can access core business apps from anywhere
   2. Buy and deploy apps in minutes
3. Spread out software costs over time

Use cases:

1. Reduce on-premises IT infrastructure and capital expenditure
2. Avoid ongoing upgrades, maintenance, and patching
3. Run applications with minimal input
4. Manage websites, marketing, sales, operations
5. Gain resilience and business continuity of the cloud provider
6. SIP – SaaS integration platforms

Concerns:

1. Data ownership and data safety
2. Third-party maintains business-critical data
3. Requires good internet connection

**Deployment models**

Deployment models indicate – where the infrastructure resides, who owns and manages it, how cloud resources and services are made available to users

**Public Cloud**

Users get access to Servers, storage, network, security, and applications delivered by cloud services providers

Users rent space on servers from the providers, no owned resources

Public cloud = cost savings

Very scalable

User has no control over environment

Service providers: AWS, Azure, IBM Cloud, Google Cloud, Alibaba Cloud

Characteristics:

1. Virtualize multi-tenant architecture enabling tenants or users to share computing resources
2. Services are NOT dedicated for single-tenant use
3. Resources distributed on as-needed basis

Benefits:

1. On-demand resources
2. Economies of scale
3. Highly reliable

Concerns:

1. Security – data loss, breach, account hijacking, etc.
2. Data sovereignty compliance – critical for companies to stay compliant as data travels across borders

Use Cases:

1. Reduce time to market for products
2. Businesses with fluctuating capacity and resourcing needs
3. Build secondary infrastructures for disaster recovery, data protection, and business continuity
4. Cloud storage and data management services for greater accessibility, easy distribution, and backing up data
5. IT depts outsourcing the management of less critical and standardized business platforms and applications to public cloud providers

**Private Cloud**

Definition - cloud infrastructure provisioned for exclusive use by a single organization comprising multiple consumers, such as the business units within the organization. It may be owned, managed, and operated by the organization, a third party, or some combination of them, and it may exist on or off premises.

Internal or External implementation

Internal – on-premises, owned and managed by organization

External – owned, managed, and operated by service provider

External cloud – Virtual Private Cloud (VCP), private cloud virtually established within a shared public cloud

Best of both worlds, all benefits of cloud services, without the concerns of public facing clouds

Public benefits:

-Dynamic scalability

-Cost efficiency

-Self service

Private benefits:

-Controlled Access

-Security

-Compliance

-Controlled internal IT

-Reduced costs

-Better scalability

-Greater agility

Common use cases:

1. Highly sensitive data
2. Modernize and unify legacy applications
3. Integrate data services from existing applications
4. Application portability, no compromise to security or compliance
5. Full control over critical security issues

**Hybrid Cloud**

Connects an organization’s on-premise private cloud and third-party public cloud into a single flexible infrastructure

-Flexibility to choose optimal cloud

-Workloads move freely

-Choice of security & regulation features

-Can leverage both clouds for same workload with proper integration and orchestration

-“Cloud bursting” – leverage additional public cloud capacity to accommodate a spike in demand for a private cloud application

Three tenants:

1. Interoperable – public and private clouds understand each other’s APIs, configuration, data formats, authentication & authorization
2. Scalable – Private clouds can leverage public cloud capacity
3. Portable – move applications & data between on-premise, cloud systems, & cloud service providers

Types of hybrid clouds

1. Hybrid monocloud – one cloud provider
2. Hybrid multicloud – can be deployed on any public infrastructure
3. Composite multicloud – Greater flexibility, allows moving components across the clouds

Benefits:

1. Security and compliance
2. Scalability and resilience
3. Resource optimization
4. Cost-saving
5. Can deploy highly regulated or sensitive workloads in a private cloud, while running less sensitive workloads in a public cloud

Use Cases:

1. SaaS integration
2. Data & AI integration
3. Enhancing legacy apps
4. VMware migration

**Deployment Model Summary**

* Deployment models indicate where the infrastructure resides, who owns and manages it, and how cloud resources and services are made available to users. There are three main deployment models available on the cloud—Public, Private, and Hybrid.
* In the Public cloud model, the service provider owns, manages, provisions, and maintains the physical infrastructure such as data centers, servers, networking equipment, and storage, with users accessing virtualized compute, networking and storage resources as services.
* In the Private cloud model, the provider provisions the cloud infrastructure for exclusive use by a single organization. The private cloud infrastructure can be internal to the organization and run or on-premises. Or, it can be on a public cloud, as in case of Virtual Private Clouds (VPC) and be owned, managed, and operated by the cloud provider.
* In the Hybrid cloud model, an organization’s on-premise private cloud and third-party, public cloud is connected as a single, flexible infrastructure leveraging the features and benefits of both Public and Private clouds.

**Week 3 - Cloud Infrastructure**

After choosing the cloud service model and the cloud type offered by vendors, customers

need to plan the infrastructure architecture.

--Availability Zones (AZ)

-Have their own power, colling, and network resources

- Improves fault tolerance, decreases latency, etc

-Very high bandwidth network connectivity

Infrastructure looks like a cloud data center

Computing resources – virtual servers (VMs), bare metal servers (Physical), “Serverless” (Abstraction layer on top of VMs)

Storage – decide how important data is when selecting storage options

-Block – traditional data centers, often struggle with scale, performance, and distributed characteristics of cloud

-File – same as block storage

-Object – most common within the cloud

Networking – Routers, switches

-Software defined networking (SDN) – virtualized networking

Public and Private interfaces

-Need IP addresses and subnets

-More important to configure network traffic and users. Use security groups, Access Control Lists (ACLs), VLANs, VPCs, VPNs

-Virtualized firewalls, load balancers, gateways, and traffic analyzers

-Content delivery networks (CDNs)

**Virtualization and Virtual Machines**

Process of software based version of something

Hypervisor – software that runs above the host

* 1. Type 1 (bare metal)
     + Installed directly on top of the host (bare metal hypervisor)
     + Most secure
     + Ex: Vmware esxi, opensource kvm, etc.
  2. Type 2 (hosted)
     + Layer of Host OS sits between physical and the hypervisor
     + Higher latency than type 1
  3. VM – software base PC, can run multiple VMs on a hypervisor
     + Run different OS on each VM
     + Extremely portable, can move from one hypervisor to another

Benefits:

1. Cost savings
2. Agility and Speed
3. Lowers downtime
   1. When one host goes out, you can move VMs over to another hypervisor

**Types of Virtual Machines**

Virtual servers, virtual instances

Variety of config and deployment options

Choice between multi and single tenant

Choice of billing types

Shared/public cloud VMs – provisioned on demand with pre-defined sizes and configs

Transient/Spot VMs – Take advantage of unused capacity in a cloud data center, cloud provider and choose to decommission them at any time. Really good for non-production workloads/testing/big data workloads

Reserved virtual server instances – Reserve capacity and guarantee resources for future deployments. Terms are in years

Dedicated hosts – single tenant isolation, specify data center and pod. Allows for maximum control. Used for meeting compliance and regulatory requirements or licensing terms

**Bare Metal Servers**

Single-tenant, dedicated server

Cloud provider managers server up to the OS

Customer is responsible for administering and managing everything else

Pre-configured by the cloud provider or custom-configured as per customer specs

-Processors

-Ram

-OS

-Specialized components

-Can install their own hypervisors

-Can add GPUs as well

Characteristics

-Take longer to provision, minutes to hours

-More expensive

-Only offered by some cloud providers

-Fully customizable/demanding environments

-Dedicated long term usage

-high performance computing

-Highly secure/isolated environments

Exampe workloads:  
- High performance compute (HPC)

-Big data analytics

-GPU intensive solutions

-ERP CRM

-AI Deep learning

-Virtualization

Comparisons:  
Bare metal:

* 1. Better for CPU and I/O intensive workloads
  2. Excel with highest performance and security
  3. Satisfy strict compliance requirements
  4. Offer complete flexibility, control and transparency
  5. Added management and op overhead

Virtual servers:

1. Rapidly provisioned
2. Provide an elastic & scalable environment
3. Low cost to use
4. Limited in throughput and performance since they share hardware

**Secure Networking in Cloud**

Cloud – Logical instances vs physical devices on premise

-NICs vs vNICs

-Network functions as a service vs networking equipment in physical racks

Define size of the network using IP address range

-VPC

--Divided into subnets

--Offer customers the security of private clouds

--Using subnets allows users to deploy enterprise applications using the same multi-tier concepts used in on premises environments

--All subnets are secured behind ACLs and Security Groups

Once subnet is built, now time to add:

-Virtual server infrastructures (VSIs)

--Web access VSI

--Applications tier VSI

--Backend Database VSI

---Place each set of VSIs into their own security groups

---Public gateway added to the Subnet

--Application responsiveness ensure using load balancers

-Storage

-- Building a cloud network entails creating a set of logical constructs that deliver networking

functionality that is akin to the data center networks that all IT professionals have come

to rely on for securing their environments and ensuring high performing business applications.

**Containers**

Executable unit of software in which application code is packaged, along with it’s libraries and dependencies, in common ways so that is can be run anywhere, including desktop, traditional IT, or the cloud.

Containers are small, fast, and portable, and unlike virtual machines, they do not need to include a guest OS in every instance and can, instead, simply leverage the features and resources of the host OS.

Example:  
Created a Node.JS application

First: VMs

--HW

--Host OS

--Hypervisor (allows up to spin up VMs)

--Linux VM

---Guest OS

---Libraries

Minimum 400mb of resources per VM

Developed application of macbook, but ran into compatibility issues when running on linux VM

Second: Containers

-Manifest – Something to describe the container (Ex: docker file)

-Create image of the application

-End up with the container that runs the application

--Host OS

--Instead of a hypervisor, we’ll have a RunTime Engine

--Deploy containers

---No guest OS, container already has libs/deps

--Uses much less resources

Now let’s add a third party service, Node.JS application now needs to access this service.

Only need to deploy 1 copy of the new application along with the already spun up containers.

--This copy becomes a shared resource of all containers in this runtime engine

**Basics of Storage on Cloud**

Save data and files in the cloud

Certain storage must be attached to a compute node, others can be directly accessed through public internet

Scale capacity as needed

Cost based on gig usage

Direct attach storage (local storage):

1. Presented directly to a cloud based server
2. Fast storage
3. Normally used only for the OS
4. Ephemeral (temporary)
5. Not-shared
6. Non-resilient even with RAID storage techniques

File storage:

1. NFS Storage – Network file system
2. Commonplace, but slower
3. Lower cost
4. Attach to multiple servers
5. Simple, works well for organizations

Block Storage:

1. Use high speed fiber connections, much faster
2. More reliable
3. Better for databases
4. Stored in volumes, mounted to compute node (1 at a time)
5. IOPS – Input/Output Operations per second
   1. How quickly data can be written/read
6. Persistance – What happens to the storage once the attached compute node is terminated
   1. The data will persist, so it can be attached to a new node unless otherwise configured
7. Snapshot (both block and file storage)
   1. Fast to create
   2. Meta data
   3. Don’t require downtime
   4. Record only the “changes” to the data
   5. Cannot be used to recover individual files

Object Storage

1. Not attached to compute node, accessed via an API
2. Least expensive
3. Slowest speeds
4. Infinite in size
   1. Pay for what you use as it grows

**File Storage**

Must be attached to a compute node before data can be stored

Less expensive

More resilient to failure

Provision larger amounts of files

More secure, encryption in transit and at rest

File storage is mounted on compute nodes via ethernet networks

-network dedicated for storage

-network attached storage or network file storage (NFS)

-Speeds varies based on network traffic

-Used for workloads where consistent network speed is not required

-Can be mounted onto more than one compute node

-Common workloads:

1. Departmental file share

2. Repository

3. “landing zone” for incoming files

-Low cost DB storage, but not very fast

-Low IOPS

**Block Storage**

-Breaks files into chunks/blocks

-Must be attached to a compute node before use

-Mounted from remote storage appliances

-Extremely resilient to failure

-Data is more secure

-Mounted as a volume using dedicated network of fiber optics

-More expensive, but much faster

-Perfect for workloads requiring low-latency

--Databases and mail servers

--Not suitable for shared storage between multiple servers

-Might be able to specify and adjust the IOPS characteristics of your storage depending on the service provider

Similarities for file and block storage:

1. Taken from appliances which are maintained by the service provider
2. Highly available and resilient
3. Often includes data encryption at rest and in transit

Differences:

1. File
   1. NFS
   2. speeds vary based on load
   3. Attach to multiple nodes at once
   4. Good for file shares where:
      1. Fast connectivity is not required
      2. Cost is a factor
2. Block
   1. Attached via fiber
   2. consistent speeds
   3. Only attach to one node at a time
   4. Good for applications that need:
      1. Consistent fast access to disk

--Remember to consider the workload IOPS requirements for both storage types

**Object Storage**

Do not connect to a compute node to use it

Provision object storage instance and use an API to upload and download data

Less expensive than other cloud storage

Effectively infinite storage, as more storage is needed you will just get more available and pay for more

Object storage uses buckets, which is essentially used the same way as a folder and they can be named

-Different buckets for different object types

-Cannot place buckets within other buckets

-Each bucket has it’s own metadata

--Info on how to locate and access the object and the time at which the data was stored or last accessed

Don’t need to provide size information when creating a bucket

Can add data as slowly or quickly as you want

Highly available

Resilience options depending on the requirements

Use cases:

1. Text files
2. Audio files
3. Video files
4. IoT data
5. VM images
6. Data archives
7. Pretty much any data which is static and where fast read and write speeds are not necessary

Object storage not suitable for OS, DBs, or changing content

**Object storage – Tiers and APIs**

Buckets have tiers/classes associated with them

Standard tier – Store objects that are frequently accessed, much more expensive (per gig)

Vault/Archive Tier – store objects that are accessed once or twice a month, lower storage cost

Cold Vault Tier – Store data that is typically accessed once or twice per year, costs just a fraction of a penny per gig per month

Automatic archiving rules – Move data to a cheaper storage tier depending on how long it’s been since last accessed

Object Storage Speed

1. No IOPS options, slower
2. Slower than file or block storage
3. Data in ‘cold vault’ buckets can take hours for retrieval
4. Not good if quick access is needed often

Costs

1. Priced per gig
2. Other costs related to retrieval of the data
3. Higher access costs for cold vault tiers

Good practice to ensure data is stored in the correct tier based on frequency of access

API

Access object storage through APIs

Standard for access is the S3 API

-Offered by AWS

-Other providers offer S3 compatible APIs as well

--HTTP Based RESTful service

--GET/PUT

Backup Solutions

--Effective solution for backup and disaster recovery

--Replacement for offsite backups

-- More efficient than tape backups (requires physical loading) for geographic redundancy

**CDN – Content Delivery Networks**

Distributed server network that delivers temp stored or cached content to users based on geographic location

Stores cached content in distributed locations and reduces distance between website visitors and the servers

Service that accelerates content delivery, faster website

Reduction of traffic/load at the main server

-Main environment gets increase in uptime

-Increase in security through obscurity

**Hybrid multi-cloud**

Embraces a mix of cloud service providers

Ex: Mail service from one cloud, platform (CRM) services from another, and infrastructure from a third

Use cases:

Cloud Scaling – provision and deprovision resources as needed

Composite cloud – applications spread across multiple cloud services

--Web UI, Billing APIs, Reward framework

--Company can move certain components into a cloud service (ex: web and billing, leaving rewards on their own premises)

--Able to take advantage of scaling at the global level by utilizing cloud data centers across the globe

Modernize – Moving certain aspects off prem, into the cloud. Ex: mobile backend in the cloud.

--Mobile backend can now communicate with on prem services

Data + AI

--Ex: Recommendation feature for users, based on planned/unplanned maintenance using legacy data

\*\*\*Another reason for multi-cloud strategy is to prevent “lock in” to a specific vendors cloud strategy

**Microservices**

Single application composed of many, loosely coupled, and independently deployable smaller components or services

Microservices typically have their own stack running on their own containers

Communicate over APIs, event streaming, and message brokers

--Can be easier and quicker to update or modify services

--Multiple devs working independently

--different stacks and runtime environments

--Independent scaling

Examples of microservices: search, recommendations, reviews, product catalogs

--They work together on a cloud platform even though they were developed independently

Service discovery – creates a roadmap for microservices to communicate (Via an API)

**Serverless Computing**

Offloads responsibility for common infrastructure management tasks such as: Scaling, scheduling, patching, and provisioning. Allows devs to focus more on business logic

--Serverless doesn’t mean “no servers.” Just means “server management” is removed from the users.

--Serverless computing environment allocates resources as needed for the apps.

Key attributes:

--No provisioning of servers & runtimes

--Runs code on-demand, scaling as needed

--Enables end users to only pay for services being used “pay only when invoked and used”

---Not paying for idle time

Serverless:

--Abstracts infrastructure away from developers

--Code executed as individual functions

--No prior execution context is required

Serverless Computing services:

--IBM Cloud Functions

--AWS Lambda

--Microsoft Azure Functions

To figure out if serverless is the right fit for you:

--Evaluate application characteristics

--Ensure app is aligned to serverless architecture patterns

Applications that qualify for serverless:

--Short running, stateless functions

--Seasonal workloads

--Production volumetric data

--Event-based processing/Asych requests

--Stateless microservices

Use Cases:

--Data and event processing

--IoT

--Microservices (most common use case)

--Mobile backends

Well suited to working with structured text, audio, images, videos

Other tasks:

1. Data enrichment
2. Transformation
3. Validation and cleansing
4. PDF Processing
5. Audio normalization
6. Thumbnail generation
7. Video transcoding
8. Data search and processing
9. Genome processing

Data stream tasks:

1. Business
2. IoT sensor data
3. Log Data
4. Financial market data

Challenges:

1. Long-running process workloads
2. Vendor lock-in
   1. Authentication
   2. Scaling
   3. Monitoring
   4. Config management
3. Start up from Zero to serve a new request (delays)

**Cloud Native Applications**

Application developed only to work in the cloud environment, or an existing app that has been refactored/reconfigured with cloud native principles

A cloud native application consists of microservices working together as a whole to comprise an

application, yet each can be independently scaled and iterated through automation and

orchestration processes.

These microservices are often packaged in containers

Enables frequent, iterative development

Development Principles:

1. Follow microservices architecture by breaking apps down to single function microservices
2. Rely on containers for scalability, portability, etc.
3. Adopt agile methods

Cloud native apps apply to hybrid and multi cloud situations

Layers:

1. Cloud/Infrastructure
   1. Public/private
2. Scheduling + orchestration
   1. Kubernetes
   2. Istio
   3. Knative
3. Application + Data services
   1. Backing services
   2. Integrate app code with existing services
4. Application runtime
   1. Middleware
5. Application code
   1. Cloud native apps – application code
   2. Designed, and built differently than conventional code

Benefits:

1. Innovation
2. Agility
3. Commoditization

Use cases for when to build cloud native apps:

1. \*everything\*
2. Standardized logging/events/catalog

**DevOps on the cloud**

Dev teams:

1. Design software
2. Develop “”
3. Deliver “”
4. Run “”

Ops Teams:

1. Monitoring
2. Predicting Failure
3. Managing Environment
4. Fixing issues

DevOps – collaborative approach that allows multiple stakeholders to collaborate:

1. Business owners
2. Development
3. Operations
4. Quality Assurance

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.

Using the DevOps approach:

1. Devs can produce software in short iterations
2. Continuous delivery schedule of new features and bug fixes in rapid cycles
3. Businesses can seize market opportunities
4. Accelerated customer feedback into products

DevOps Process:

1. Continuous delivery
   1. Small, well designed, high quality
2. Continuous Integration
   1. Creating packaged builds as immutable images
3. Continuous deployment
   1. Progressing new package through deployment lifecycles asap
4. Continuous monitoring
   1. Understand performance and availability
5. Delivery pipeline
   1. Ideation
   2. Coding
   3. Building
   4. Deploying
   5. Managing
   6. Continuous improvement

DevOps and Cloud

DevOps tools, practices, and processes help to tackle the risks and challenges of utilizing cloud platforms

Core capabilities:

1. Automated provisioning and installation
2. CI/CD, automated deployment pipeline
3. Define how people work together
4. Test in low-cost, production-like environments
5. Recover from disasters by rebuilding systems quickly and reliably

**Application Modernization (App Mod)**

Legacy apps

1. “siloed” systems
2. Difficult to update
3. Expensive to maintain

Modernized apps

1. Accelerate digital transformation
2. Leverage new tech & services
3. Respond faster to change

3 main ingredients to modernization

1. Architecture
   1. Monoliths -> Service oriented architecture (SOA) -> Microservices
2. Infrastructure
   1. Physical Servers -> VM -> Cloud (public or private)
3. Delivery
   1. Waterfall -> Agile -> DevOps

Basically can’t have one of these without the other, all tied together

**What is Cloud Security – Part 1**

Shared responsibility between “you” and the cloud provider

PaaS

-“You” are responsible for securing applications, workload, and the data

-Cloud provider responsible for managing security of the platform

IaaS

-Cloud provider manages hypervisor and below

Saas

-Cloud provider manages everything, “you” only worry about the data

Architecture

-Data at rest encryption

-Key management

-Confidential vs sensitive data

-More control == more responsibility

-Protect data in memory (using hardware based tech)

-Protect data in motion

-Bring your own keys (BYOK)

-Keep your own keys (KYOK)

-Data access on a need-to-know basis

-Scan apps (App Sec approach) before deployment

-Scan container images before deployment for vulnerabilities (container sec)

-Security is easier to build in for a cloud native environment

-Identity checks for users

--authorized users only

-Network security

--firewall, network Access Control, DDOS protection

\*\*Manage access, protect data

**What is Cloud Security – Part 2**

Continuous security monitoring

-Gain insight about your

1. posture

2. compliance

3. threats

-Using logs

-remediate containers

DevOps

Should be “SecDevOps” – embed security

Secure Design – planning questions: what kind of data? What level of classification? Container based? Migrating workloads?

Secure Build, manage security

Closed loop for easier/quicker remediation

**Identity and Access Management**

Cloud security concerns:

1. Data loss and leakage
2. Unauthorized access
3. Insecure interfaces and APIs

Identity and Access Management

1. First line of defense
2. Authenticate and authorize users
3. Provide user-specific access to resources

Main types of users:

A comprehensive security strategy needs to encompass the security needs of a wide audience

1. Organizational users
2. Internet and social-based
3. Third-party business partner organizations
4. Vendors

3 main types:

1. Administrative
   1. Cloud admins, operators, managers
      1. Roles that typically create, update, and delete application and instances, and also need insight into their team members activities
   2. An attacker on an admin account could:
      1. Steal data from DBs
      2. Deploy malicious apps
      3. Deface or destroy existing apps
2. Developers
   1. Cloud app devs, platform devs, app publishers
      1. Read sensitive info
      2. Create/update/delete applications
3. Application Users
   1. User of cloud hosted applications

Key components of identity and access management:

1. Cloud directory
2. Social login
3. Enterprise identity provider
4. Cloud identity provider
5. API keys
6. Multifactor authentication – used to combat identity theft by adding an additional level of authentication for application users
   1. Single use PW
   2. Pin
   3. Certificates
   4. Tokens
   5. Risk based – changes in user location/activity/preferences
7. Cloud directory services – used to securely manage user profiles and their associated credentials and PW policy inside a cloud environment
   1. Apps hosted on the cloud do not need to use their own user repo
8. Reporting – helps provide a user-centric view of access to resources or a resource-centric view of access by users
   1. Which users have access to which resources?
   2. Which access is being abused?
9. Audit and compliance – critical service within identity and access management framework, both for cloud provider, and cloud consumer
   1. Validate implemented controls against:
      1. Security policy
      2. Industry compliance
      3. Risk policies
      4. Report deviations
10. User and service access management
    1. enables cloud app/service owners to provision and de-provision:
       1. Customer profiles
       2. Partner profiles
       3. Vendor profiles
    2. Streamline access control based on:
       1. Role
       2. Organization
       3. Access policies
11. Mitigating risks
    1. Provisioning users by specifying roles on resources for each user
    2. PW policies that control the usage of special characters, minimum PW lengths, similar settings
    3. Multifactor authentication
    4. Immediate de-provisioning when users leave or change roles
12. Access groups – 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
    1. Create access groups
    2. Add users to groups
    3. Manage access for existing users
13. Access policies – define how users, service IDs, and access groups in the account are given permission to access account resources
    1. Policies include:
       1. Subject – users, services IDs, or access groups
       2. Target – resource or service offering
       3. Role – defines actions allowed on the target of the policy
    2. Benefits
       1. Streamline access assignment process vs assigning individual user access
       2. Reduce number of policies

**Cloud Encryption**

Plays a key role on cloud, often last line of defense in a layered security model.

1. Encrypts data
2. Data access control
3. Key management
4. Certificate management

Definition – scrambling data in a way that makes it illegible

Encryption algorithm – defines the rules by which data will be transformed

Decryption key – defines how encrypted data will be transformed back to legible data

Encryption ensures:

1. Only authorized users have access to sensitive data
2. When accessed without authorization, data is unreadable and meaningless

Cloud encryption services:

1. Can be limited to encryption of data or
2. End-to-end encryption of all data uploaded

Data protection states:

1. At rest
   1. Protects stored data
   2. Options:
      1. Block and file storage
      2. Built-in for object storage
      3. Database encryption
2. In Transit
   1. Protects data while transmitted
   2. Includes encrypting before transmission
   3. Authenticates endpoints
   4. Decrypts data on arrival
   5. Options:
      1. Secure Sockets Layer (SSL)
      2. Transport Layer Security (TLS)
3. In Use
   1. Protects data in memory
   2. Allows computations to be performed on encrypted text without decryption

Encryption can be server or client side  
Server-side

1. Create and manage own keys (customer supplied) or
2. Generate and manage keys on the cloud (customer managed)

Client-side

1. Occurs before data is sent to cloud
2. Cloud providers cannot decrypt hosted data

\*\*\*There is aneed to implement a singular data protection strategy across an enterprise’ on-premise, hybrid, and multi-cloud deployments

Multi-cloud data encryption features:

1. Data access management
2. Integrated key management
3. Sophisticated encryption

Data encryption console:

1. Define and manage access policies
2. Create, rotate, and manage keys
3. Aggregate access logs

Key management

-Encryption does not eliminate security risk

1. separates the security risk from the data itself

2. keys need to be managed and protected against threats

Key management services offered by some cloud providers

1. Encrypt sensitive data at rest
2. Easily create and manage the entire lifecycle of cryptographic keys
3. Protect data from cloud service providers since keys remain in possession of the owners

Key management best practices

1. Storing encryption keys separately from the encrypted data
2. Taking key backups offsite and auditing them regularly
3. Refreshing the keys periodically
4. Implementing multi-factor authentication for both the master and recovery keys

**Cloud Monitoring Basics and Benefits**

Monitoring performance across an entire stack of apps and services can be time-consuming and draining on internal resources.

Cloud monitoring assessment

Assess data, application, and infrastructure behaviours for:

1. Performance
2. Resource allocation
3. Network availability
4. Compliance
5. Security Risks/threats

Cloud monitoring includes:

1. Strategies
2. Practices
3. Processes

Used for:

1. Analyzing
2. Tracking
3. Managing services and apps

It also serves to provide actionable insights that can help improve availability and user experience.

Cloud monitoring helps to:

1. Accelerate diagnosis and resolution of performance incidents
2. Control the cost of monitoring infrastructure
3. Mitigate the impact of abnormal situations with proactive notifications
4. Get critical Kubernetes and container insights for dynamic microservice monitoring
5. Troubleshoot application and infrastructure

Cloud monitoring solutions provide:

1. Data in real-time with round the clock monitoring of VMs, services, databases, and apps
2. Multilayer visibility into app, user, and file access behavior across all apps
3. Advanced reporting and auditing capabilities for ensuring regulatory standards
4. Large-scale performance monitoring integrations across multicloud and hybrid cloud

Cloud monitoring categories

1. Infrastructure
   1. Help identify minor and large scale failures
   2. Devs can take corrective actions
2. Database
   1. Help track processes, queries and availability of services
   2. Ensures accuracy and reliability
3. Application performance monitoring
   1. Help improve user experience
   2. Meet app and user SLAs
   3. Minimize downtime and lower operational costs

Best practices:

1. Leverage end user experience monitoring solutions
   1. Helps to improve customer experience
2. Move all aspects of infrastructure under one monitoring platform
   1. Complete visibility
3. Use monitoring tools that help track usage and cost
4. Increase cloud monitoring automation
   1. Help gain operational efficiency
5. Simulate outages and breach scenarios
   1. Evaluate tool efficiency

**Career opportunities**

1. Cloud software engineer – software dev lifecycle
   1. Writing, testing, maintaining
   2. App front and back end
   3. Platforms and systems
   4. Experience with at least one major cloud provider
   5. Knowledge of data structures, distributed systems, OS, algorithms
   6. Experience with databases
   7. Proficiency in web app development – python, javascript, java, html, css
2. Cloud integration specialists
   1. Integrate new cloud services, applications and infrastructure with existing systems and services
   2. Assess implications and trade-offs between different solutions
   3. Optimize integration
   4. Optimize user experience
   5. Optimize performance standards to meet Service Level Agreements (SLAs)
3. Cloud Data engineers
   1. Design, develop, deploy data pipelines and services
   2. Integrate new data management technologies and software engineering tools into existing infrastructure
   3. Recommend automated integration of disparate data sets
   4. Collaborate with Data Scientists to create predictive models & PoCs
   5. Promote best practices for consumption and understanding of data
   6. Improve efficiency by introducing new engineering processes and tools
4. Cloud security engineers
   1. Provide expertise to protect organizations systems and application data (confidentiality, integrity, availability)
   2. Determine security requirements
   3. Plan, implement, and test security systems
   4. Recommend innovative technologies to enhance security of cloud based environments
   5. In-depth knowledge of cloud platforms and services, software design patterns and DevOps tools and methodologies
5. Cloud DevOps Engineers
   1. Collaborate with dev and ops team to create reliable and rapid release pipelines for software and updates
   2. Create custom automation tools
   3. Build and maintain configuration and deployment frameworks
   4. Track design bugs and automate debugging process
   5. Maintain and deploy web-based applications
   6. Monitor security systems
   7. Measure performance against expected business outcomes
   8. Containerization expertise
6. Cloud solutions architects
   1. Translate business requirements into application architecture and design
   2. In-depth understanding of cloud platforms and services
   3. In-depth understanding of software design patterns
   4. Knowledge of DevOps tools and Methodologies
   5. Good understanding of networking
   6. High level of understanding of security concepts
   7. Collaborate with cloud developers, networking specialists, security engineers, integration specialists, and devops engineers