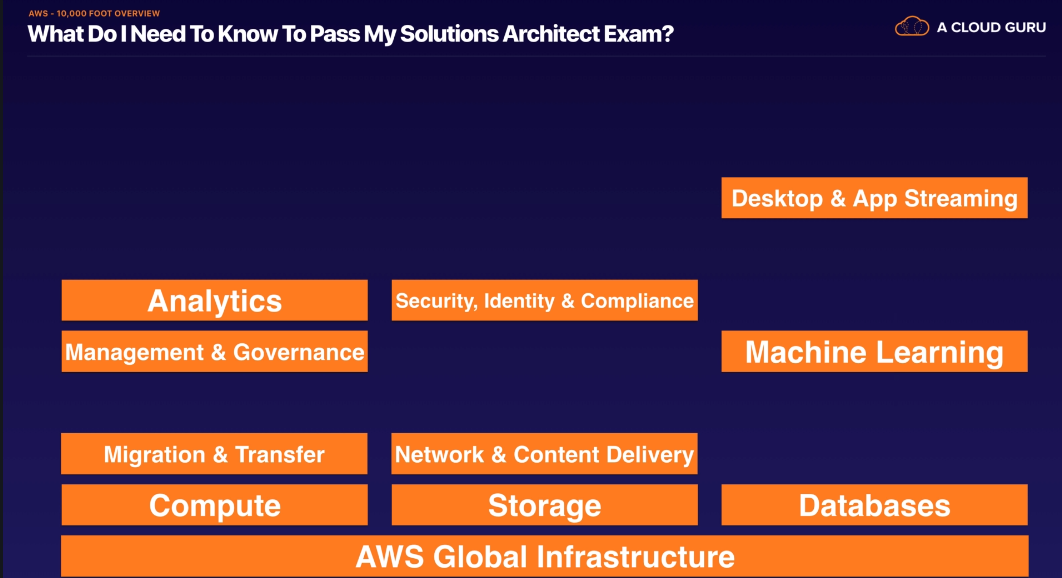
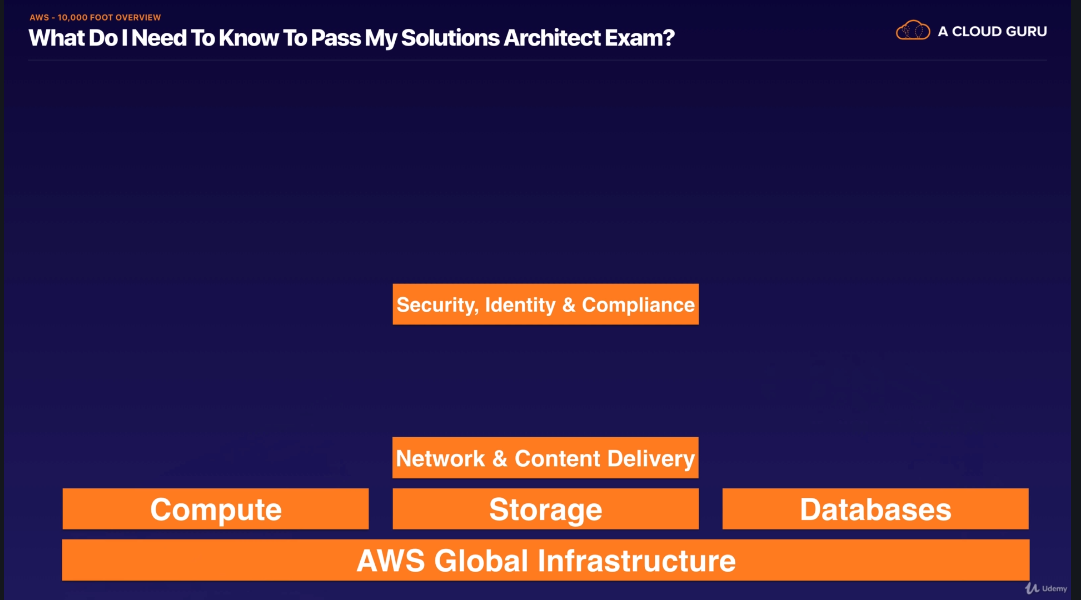
**AWS**

* **10,000 ft Exam Tips**
  + Edge Locations
    - Endpoints for AWS which are used for caching content.
    - Typically consists of
      * CloudFront, Amazon’s CDN (Content Delivery Network
    - There are many more Edge Locations than regions
      * Currently about 150 edge locations
  + Understand difference between Availability Zone and an Edge Location
    - # of Edge Locations > # of Availability Zones > # of Regions
    - **A region** is a physical location in the world which consists of two or more Availability Zones (AZ’s)
      * Is a separate geographic area
      * Each region has multiple, isolated locations known as Availability Zones
    - An **AZ** (**Availability Zone)** is one or more discrete data centers, each with redundant power, networking and connectivity, housed in a separate facilities
      * Distinct location within a region
      * Each Region has at least two AZs
    - **Edge Locations** are endpoints for AWS which are used for caching content. Typically this consists of cloudfront, Amazon’s CDN
  + VPC is part of Networking Services for AWS
    - It is a provision of a server built for availability and security
  + 
  + \*\*\* MOST IMPORTANT \*\*\*\*
  + 
* Free tier refers to the limited use of AWS
  + Compute (2nd definition is official AWS, 1st is simplified)
    - EC2 (Elastic Cloud Compute)
      * Some processors are included in free tier
      * Basically, a virtual computer, very similar to the desktop or laptop computer you use at home, and commonly referred to as an ***instance***
      * Web service that provide resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers
    - ELB (Elastic Load Balancing)
      * A tool that distributes incoming web traffic (i.e. visitors to a web site) and equally across multiple EC2 instances that are running a web site.
        + Helps prevent one server from being overloaded while another server can handle more visitors
      * ELB automatically distributes incoming application traffic across multiple Amazon EC2 instances. It enables fault-tolerance in applications seamlessly providing the required amount of load-balancing capacity needed to route application traffic
    - Lambda (Serverless Computing)
      * Serverless computing essentially, allowing code to be run without setup and maintenance of actual servers
      * Lambda lets you run code without having to provision or manage servers. You’re only charged for the compute time you consume. There is no change when your code is not running. With Lambda you can run code for virtually any type of application or backend service. All with zero admin. You just upload your code and Lambda takes care of everything required to run and scale with high availability
    - Asdf
* VPC notes
  + Region and Availability Zones
    - Regions throughout world
      * Each region is completely independent
      * Regions are completely isolated from others to achieve greatest possible fault tolerance and stability
    - Each region houses multiple availability zones
    - Ec2 allows you to create instances and data in multiple locations.
      * Resources are not replicated across regions unless you do so specifically, default is not replicated
      * When you launch an instance, you must select an AMI that’s in the same Region
      * If AMI is in another region you can copy over the AMI to the region you are using
      * Charge for data transfer between regions
    - Availability Zones
      * Can be selected upon creation or launch of an instance
      * Can design so that if instance fails another instance takes over in another availability zone
      * Can use Elastic IP addresses to mask failure of an instance by rapidly remapping the address to an instance in another zone
      * When you launch an instance, select a region close to your customers. Can protect from failure by using separate availability zones
      * Must select regional endpoints
    - Internet Gateways
* IAM
  + Identity Access Management
  + **Exam TIPS**
    - IAM is universal. It does not apply to regions at this time
      * Created user, role or group are created globally
    - The “Root Account” is simply the account created when first setup your AWS account. It has complete Admin access
    - New Users have NO permissions when first created
    - New Users are assigned Access Key ID & Secret Access Keys when first created
      * **Access Key ID & Secret Access Keys are not the same as a password.** If you lose them, you have to regenerate them. So, save them in a secure location.
    - Always set up Multifactor Authentication on your root account
      * Simple, just need a smart phone and google authenticator
    - You can create and customize your own password rotation policies
  + Key Features
    - Centralised control of your AWS Account
    - Shared Access to your AWS Account
    - Granular Permissions
    - Identity Federation( including Active Directory, FB, LinkedIn)
    - Multifactor Auth
    - Provide Temp Access for users/devices and services when necessary
    - Allows you to set up your own password rotation policy
    - Integrates with many different AWS services
    - Supports PCI DSS Compliance
      * If taking credit card details will need to support this
  + Key Terminology
    - Users
      * End Users i.e. ppl, employees
    - Groups
      * Collection of users
      * Each user in the group will inherit the permissions of the group
    - Policies
      * Policies are made up of documents, called Policy Documents
      * Documents are in a form called JSON
      * Give permissions as to what a User/Group/Role is able to do
    - Roles
      * Create roles
      * Assign them AWS Resources
  + Identity Federation
    - AD
    - Facebook
    - LinkedIn
  + MFA
  + Allows you to set up your own password rotation policy
  + Key Terms
    - Users
      * End Users
    - Groups
      * Collection of Users
      * Each user will inherit the permissions of the group
    - Policies
      * Made up of documents
      * In JSON format
* EC2 notes
  + Instances
    - # of virtual CPUs
    - How much memory
    - How much bandwidth
  + EC2 Basics
    1. Purchasing options
       1. On-demand
       2. Reserved
       3. Spot
    2. Instance Types (processing capacity)
       1. General Purpose
       2. Compute Optimized
       3. Accelerated Computing
       4. Memory Optimized
       5. Storage Optimized
    3. EBS optimized
       1. Option for higher IOPS
    4. AMI Type (operating system)
       1. Linux
       2. Windows
    5. Data Transfer
       1. In/out of the instance
    6. Region
       1. Where the instance is provisioned in
  + Volumes
    - What are volumes?
      * Considered to be like storage or external storage or partitioned hard drives. The root will be deleted by default but any additional volume will not be. These can be attached or detached when the instance is terminated
    - Snapshots, what are they?
      * Snapshot is an image of an EBS volume that can be stored as a backup of the volume or used to create a duplicate
      * A snapshot is not an active EBS volume. Cannot attach or detach a snapshot to an EC2 instance
      * To restore a snapshot, you need to create a new EBS volume using the snapshot as its template
  + IOPS
    - Input/Output Operations Per Second
    - Simple def
      * Amount of data that can be written to or retrieved from EBS per second
    - AWS def
      * IOPS are a unit of measure representing input/output operations per second. The second operations are measure in KiB, and the underlying drive technology determines the maximum amount of data that volume type counts as a single I/O. I/O size is capped at 256 KiB for SSD volumes and 1,024 KiB for HDD volumes because SSD volumes handle small or random I/O much more efficiently than HDD volumes
    - More IOPS mean better volume performance
    - What determines the amount of IOPS?
      * EBS volume size. The larger the storage size (in GiB), the more IOPS the volume has
  + Security Groups
    - Are found on the instance level as opposed to the subnet level.
    - The way allow/deny works is different than NACL
    - Simplified def
      * Security Groups are very similar to NACLs; they allow or deny traffic. However, security groups are found on the instance level (as opposed to the subnet level). In addition, the way allow/deny rules work are different from NACLs
    - AWS def
      * Acts as a virtual firewall that controls the traffic for one or more instances. When you launch an instance you associate one or more security groups with the instance. You add rules to each security group that allow traffic to or from its associated instances. You can modify the rules for a security group at any time. New rules are automatically applied to all instances that are associated with the security group.
        + We decide whether to allow traffic to r each an instance, we evaluate all the rules form all the security groups that are associated with the instance

**S3 – Simple Storage Service**

**EXAM TIPS –**

1. Remember that S3 is **Object-based**: i.e. allows you to upload files
2. Files can be from 0 byes to 5 TB
3. There is unlimited storage
4. Files are stored in Buckets
5. Control Access to Buckets using either a **Bucket ACL** or using **Bucket Policies**
6. S3 is a universal namespace. That is, must be globally unique
   1. Ex.
7. Not suitable to install an operating system on
   1. You would want block based storage for this
   2. S3 is object based remember
8. Successful uploads will generate a HTTP 200 status
   1. Upload an object
9. You can turn on MFA Delete
   1. Prevents accidentally deleting your objects
10. Key Fundamental of S3 are:
    1. Key (This is simply the name of the object)
    2. Value (this is simply the data, made up of a sequence of bytes)
    3. Version ID (Important for versioning
    4. Metadata (Data about data you are storing)
    5. Sub resources;
       1. Access Control Lists
       2. Torrent
    6. Consistency Model
       1. Read after Write consistency for PUTS of new Objects
       2. Eventual Consistency for overwrite PUTS and DELETES (can take some time to propagate)
          1. Meaning if you look you may still see the old object but if you wait its going to be consistent so you get eventual consistency
    7. Class (S3, S3 – IA, S3 – IA (One Zone), Glacier)
       1. S3 Standard
          1. 99.99% availability
          2. 99.999999999% durability, stored redundantly across multiple devices in multiple facilities, and is designed to sustain the loss of 2 facilities concurrently
       2. S3 – IA (Infrequent Access)
          1. Less frequently accessed data but required quickly when needed
          2. Lower Fee than S3
             1. However charged a retrieval fee
       3. S3 One Zone – IA
          1. Previously called RRS (Reduced Redundancy Storage)
          2. Lower cost option for infrequently accessed data, but do not require the multiple availability zone data resilience
       4. S3 Intelligent Tiering
          1. Designed to optimize costs by automatically moving data to the most cost-effective access tier, without performance impact or operational overhead
       5. S3 Glacier
          1. S3 glacier is a secure, durable, and low cost storage class for data archiving. Retrieval time configurable from minutes to hours
       6. S3 Glacier Deep Archive
          1. S3 Glacier Deep Archive is Amazon S3’s lowest cost storage class where a retrieval time of 12 hours is acceptable.
    8. FAQs

* Simple def
  + An online, bulk storage service that you can access from almost any device
* AWS def
  + Amazon S3 has a simple web services interface that you can use to store and retrieve any amount of data, at any time, from anywhere on the web. It gives any user access to the same highly scalable, reliable, fast inexpensive data storage that Amazon uses to run its own global network of web sites. The services aims to maximize benefits of scale and to pass those benefits on to user
* Basics
  + Components and Structure
    - Basics
      * S3 = Simple Storage Service
      * It is AWS’s primary storage service
      * You can store any type of file in S3
    - Buckets
      * Root level “Folders” you create in S3 are referred to as buckets
      * Any “Subfolder” you create in a bucket is referred to as a folder
    - Objects
      * Files stored in a bucket are referred to as objects
    - Regions
      * When you create a bucket, you must select a specific region for it to exist. This means that any data you upload to the S3 bucket will be physically located in a data center in that region
      * Best practice to to select the region that is physically closest to you, to reduce transfer latency
      * If you are serving files to a customer based in a certain area of the world, create the bucket in a region closest to them (to reduce latency)
  + Costs / Pricing
    - Free tier available
    - How are you charged?
      * Storage Cost
        + Applies to data at rest in S3
        + Per GB used
        + Price per GB varies based on region and storage class
      * Request Pricing
        + PUT
        + COPY
        + POST
        + LIST
        + GET
        + Lifecycle Transition Request
        + Data Retrieval
        + Data Archival
        + Data Restoration
* Buckets & Objects
  + Creating an S3 Bucket:
    - Bucket Names must follow a set of rules:
      * Bucket Names must be unique across all of AWS
      * Bucket Names must be 3 to 63 characters in length
      * Bucket names can only contain lowercase letters, numbers, and hyphens
      * Bucket names must not be formatted as an IP address (e.g. 192.168.5.4)
    - Select a region
      * Best practice is closest to your users
    - Uploading an object to a bucket
      * Create a bucket
      * Click into
      * Upload
    - Create a Folder in a bucket
      * Navigate into a bucket
      * Create folder
      * Name folder
    - Properties
      * Bucket Level
        + General Info
        + Permissions
        + Static Web Hosting
        + Logging
        + Events
        + Versioning
        + Lifecycle
        + Cross-Region Replication
        + Tags
        + Requester Pays
        + Transfer Acceleration
      * Folder Level Properties
        + General Info
        + Details
      * Object Level Properties
        + General Info
        + Details
        + Permissions
        + Metadata
* Storage Class
  + Represents the classification assigned to each object in S3
  + Available Storage Classes include
    - Standard
      * Designed for general, all purpose storage
      * Default storage option
      * 99.999999999999% object durability
      * 99.99% object availability
      * Most expensive storage class
    - Intelligent-Tiering
      * Designed for objects with changing or unknown access patterns. S3 monitors access patterns of the object and moves the ones that have not been accessed to the infrequent access tier
      * 99.999999999999% object durability
      * 99.90% object availability
      * Less expensive than standard storage classes
    - Standard Infrequent Access (S3 Standard-IA)
      * Designed for objects you do not access frequently, but must be immediately available when accessed
      * 99.999999999999% object durability
      * 99.90% object availability
      * Less expensive than the standard storage classes
    - One Zone-Infrequent Access (S3 One Zone-IA)
      * Designed for non-critical, reproducible objects
      * 99.999999999999% object durability
      * 99.50 object availability
      * Less expensive than the standard storage class and S3 standard-infrequent Access
    - Glacier
      * Designed for long-term archival storage
      * May take several hours for object stored in Glacier to be retrieved
      * 99.999999999999% object durability
      * Low cost S3 storage class (very low cost)
    - Gracier Deep Archive
      * Designed for long-term archival storage
      * May take several hours for object stored in Glacier to be retrieved
      * 99.999999999999% object durability
      * Cheapest S3 storage class (very low cost)
    - Reduced Redundancy (not recommended)
  + Each Storage Class has varying attributes that dictate things like
    - Storage Cost
    - Object Availability - % over a one-year time period that a file stored in S3 will be accessible
      * For every 10,000 hours, you can expect a total of one hour for which a file may not be available to access
    - Object durability - % over a one year time period that file stored in S3 being lost in a year
      * I.e. 10,000 files stored in S3 (@ 11 nines durability) then you can expect to lose one file every 10 million years
    - Frequency of Access (to the object)
  + Each object must be assigned a storage class
    - Standard is the default
  + You can Change the storage class of an object at any time for the most part
* Setting Up / Changing Storage Class
  + By default, all new object uploaded to S3 are set to standard
  + If you want new objects to have a different storage class, you need to set things properly prior to ordering the upload process. You can do this by either
    - Selecting another storage class
    - Using object lifecycle policies
  + For the following four ( can switch the objects storage class by changing the storage class in the objects properties )
    - Standard
    - Intelligence-Tiering
    - Standard - infrequent Access
    - One Zone- infrequent Access
  + To move an object to the glacier and Glacier Deep Archive storage class
    - Need to use object lifecycle
    - The change to glacier may take one to two days to take effect
* Object Lifecycles
  + Automates moving object between different storage tiers
  + Can be used in conjunction with versioning
  + Can be applied to previous and as well as current versions
  + An object lifecycle is a set of rules that automate the migration of an object’s storage class to a different storage class, or its deletion, based on specified time intervals
    - I.E. (Scenario)
      * I have a work file that I am going to access every day for the next 30
      * After 30, may need file weekly for next 60 days
      * After 90 days from prior two, I will likely never access the file again but want to keep it just in case
    - By using a lifecycle policy, I can automate the process of changing the files storage class to meet my usage needs and keep my S3 storage cost as low as possible
    - Scenario Solution
      * Day 0-29 (1st 30 days)
        + Usage needs = Very Frequent
        + “Best Fit” Storage class = **Standard**
        + Cost = Highest Cost Tier
      * Day 30-89 (Next 60 days)
        + Usage need = Infrequent
        + “Best Fit” Storage class = **Standard Infrequent Access**
        + Cost = middle cost tier
      * Day 90 + (going forward)
        + Usage Needs = Most likely never needed
        + “Best Fit” Storage class = **Glacier**
        + Cost = lowest cost tier
  + Lifecycle Management
    - Lifecycle’s functionality is located on the bucket level
    - However a lifecycle policy can be applied to:
      * The entire bucket
      * One specific Folder within a bucket (applies to all object in that folder)
      * One specific object within a bucket
    - You can always delete a lifecycle policy or manually change the storage class back to whatever you like
* Lifecycle Permissions
  + S3 Permissions functionality can be found on the bucket and object level
    - Permission functionality can be found on both the bucket and on the object level
    - On the bucket level you can control (for each bucket individually)
      * List: Who can see the bucket name
      * Upload/Download: Object to (upload) or in the bucket (delete)
      * Permissions: Added/edit/delete/view permissions
        + NOTE\*\* Bucket level permissions are generally used for “internal” access control
    - On the Object Level, you can control: (for each object individually)
      * Open/Download
      * View Permissions
      * Edit Permissions
  + Sharing an S3 object with the worlds
    - On the object check the following permissions
      * Grantee = Everyone
      * Check Open/Download
    - Under Actions, select Make Public
    - The link under Properties is now live, and anyone that has it can directly download the object
      * \*\*NOTE\*\* To remove public access to the object either delete the permission or remove the bucket policy that provided the public access
* S3 Versioning
  + Stores all versions of an object (including all writes and even if you delete an object)
  + Is a feature that keeps track of and stores all versions of an object so that you can access and use an older version if you like
    - Including all writes and even if you delete an object
  + Great backup tool
  + Integrates with LifeCycle rules
  + Versioning comes with MFA delete capability, which uses MFA, can be used to provide an additional layer of security.
    - Requires authenticating prior to deleting objects
  + Options
    - Versioning is either on/off
    - Once it is turned on, you can only suspend versioning. Cannot be fully turned off
    - Suspended versioning only prevents versioning going forward. All previous objects with versions will remain in storage
    - Versioning can only be set at the bucket level, and applies to all object in a bucket
* Encryption
  + By default:
    - All newly created buckets are private. You can setup access control to your buckets using
      * Bucket Policies
      * Access Control Lists
    - S3 buckets can be configured to create access slogs which log all requests made to the S3 bucket. This can be sent to another bucket and even another bucket in another account.
  + Encryption in Transit
    - SSL/TLS
    - When you visit a website, HTTPS
      * Traffic is encrypted in transit
      * Nobody who is eavesdropping on this can steal the data
  + Encryption At Rest (Server Side) is achieved by
    - If someone steal hard disk drive
      * If there is no password they can read drive
    - Server Encryption
      * S3 Managed Keys – SSE – S3
      * AWS Key Management Service, Managed Key – SSE-KMS
      * Server Side Encryption With Customer Provided Keys – SSE – C
  + Client side
    - Client Side Encryption
      * Password?

**Databases – S3**

* RDS (Relational Database Service) and DynamoDB
  + RDS for SQL DB
    - Simple Def
      * RDS is an SQL database service that provides a wide range of SQL database options to selection from
      * Options
        + Amazon Aurora
        + MySQL
        + MariaDB
        + PostgreSQL
        + Oracle (several options)
        + Microsoft SQL Server (several options)
    - AWS Def
      * Amazon Relational Database Service (Amazon RDS) is a web service that makes it easier to set up, operate, and scale a relational database and manages common database admin tasks
  + DynamoDB for NoSQL DB
    - Simple Def
      * DynamoDB is a NoSQL database service. Unlik