**Table of Contents**

[Introduction 3](#_Toc12475693)

[region, an Availability Zone (AZ) and an Edge Location 3](#_Toc12475694)

[Identity Access Management 3](#_Toc12475695)

[LAB 9: Identity Access Management - IAM 4](#_Toc12475696)

[Lab 10: Create A Billing Alarm – CloudWatch 4](#_Toc12475697)

[Amazon S3 5](#_Toc12475698)

[S3 5](#_Toc12475699)

[Lab 12: Create an S3 Bucket 7](#_Toc12475700)

[Security & Encryption 7](#_Toc12475701)

[Lab 14: Version Control 8](#_Toc12475702)

[Lab 15: Lifecycle Management and Glacier 8](#_Toc12475703)

[Lab 16: Cross Region Replication 8](#_Toc12475704)

[S3 Transfer Acceleration 8](#_Toc12475705)

[CloudFront CDN 9](#_Toc12475706)

[Lab 18: Create a CloudFront Distribution 9](#_Toc12475707)

[Snowballs 10](#_Toc12475708)

[Lab 21: Snowball 10](#_Toc12475709)

[Storage Gateway 10](#_Toc12475710)

[Ec2 – Elastic compute cloud 12](#_Toc12475711)

[EC2 12](#_Toc12475712)

[Lab 25: Launch an EC2 Instance 14](#_Toc12475713)

[Lab 27: Security Groups 15](#_Toc12475714)

[EBS 15](#_Toc12475715)

[Lab 29: EBS Volumes & Snapshots 16](#_Toc12475716)

[AMI Types (EBS vs Instance Store) 16](#_Toc12475717)

[Lab 31: Encrypted Root Device Volumes\_Snapshots 17](#_Toc12475718)

[Cloudwatch 17](#_Toc12475719)

[Lab 33: CloudWatch 18](#_Toc12475720)

[Lab 34: AWS Command Line (CLI) 18](#_Toc12475721)

[Identity Access Management Roles 19](#_Toc12475722)

[Lab 36: Using Boot Strap Scripts 19](#_Toc12475723)

[Lab 37: EC2 Instance Metadata 19](#_Toc12475724)

[Lab 38: EFS (Elastic File System) 19](#_Toc12475725)

[EC2 Placement Groups 19](#_Toc12475726)

[Databases 20](#_Toc12475727)

[Databases 20](#_Toc12475728)

[Lab 42: Create RDS Instance 21](#_Toc12475729)

[RDS – Back Ups, Multi-AZ & Read Replicas 22](#_Toc12475730)

[Lab 44: RDS – Backups, Multi-AZ & Read Replicas 23](#_Toc12475731)

[DynamoDB 24](#_Toc12475732)

[Redshift 24](#_Toc12475733)

[Aurora 26](#_Toc12475734)

[Elasticache 26](#_Toc12475735)

[Route 53 27](#_Toc12475736)

[DNS 27](#_Toc12475737)

[Lab 51: Register A Domain Name 29](#_Toc12475738)

[Routing Policies Available on AWS 29](#_Toc12475739)

[VPCs 31](#_Toc12475740)

[VPC - Virtual Private Cloud 31](#_Toc12475741)

[Lab 62: Build A Custom VPC 32](#_Toc12475742)

[Lab 64: NAT Instances & NAT Gateways 32](#_Toc12475743)

[Lab 65: NACL - Network Access Control Lists 33](#_Toc12475744)

[Lab 66: Custom VPCs and ELBs 33](#_Toc12475745)

[Lab 67: VPC Flow Logs 33](#_Toc12475746)

[Bastion Host 34](#_Toc12475747)

[Direct Connect 34](#_Toc12475748)

[VPC Endpoints 35](#_Toc12475749)

[HA Architecture 35](#_Toc12475750)

[Elastic Load Balancers 35](#_Toc12475751)

[Lab 36: Load Balancers 36](#_Toc12475752)

[Advanced Load Balancer Theory 36](#_Toc12475753)

[Lab 75: Launch Configurations & Auto Scaling Groups 37](#_Toc12475754)

[HA Architecture 37](#_Toc12475755)

[Lab 77-80: Building a fault tolerant WordPress site 37](#_Toc12475756)

[Lab 82: Elastic Beanstalk 37](#_Toc12475757)

[Application 37](#_Toc12475758)

[SQS - Simple Queue Service 37](#_Toc12475759)

[SWF - Simple Work Flow Service 39](#_Toc12475760)

[SNS - Simple Notification Service 39](#_Toc12475761)

[Elastic Transcoder 40](#_Toc12475762)

[API Gateway 40](#_Toc12475763)

[Kinesis 42](#_Toc12475764)

[Web Identity Federation & Cognito 43](#_Toc12475765)

[Serverless 44](#_Toc12475766)

[Lambda 44](#_Toc12475767)

[Lab 93: Build a serverless webpage with API Gateway & Lambda 45](#_Toc12475768)

[Lab 94: Build an Alexa Skill 45](#_Toc12475769)

# Introduction

1. Certified Solutions Architect associate exam
2. Core course:
   1. Security, Identity & Compliance
      1. Identity access management
   2. Network & Content Delivery
      1. Route 53 (Amazon’s DNS service)
      2. VPC (virtual private cloud) – virtual datacenter
   3. Compute
      1. Add lambda
   4. Storage
      1. S3
   5. Database
      1. DS
      2. Dynamo DB
      3. Redshift

## region, an Availability Zone (AZ) and an Edge Location

1. Region – a physical location which consists of 2 or more Availability Zones
2. Availability Zones (AZ) - one or more discrete data centers, each with redundant power, networking and connectivity, housed in separate facilities.
3. Edge Locations are endpoints for AWS which are used for caching content.
   1. Edge locations consists of CloudFront and Amazon’s Content Delivery Network (CDN)

# Identity Access Management

1. What is IAM?
   1. Allow you to manage users and their level of access management to the AWS console.
   2. IAM is globally available and not specified to region
   3. 使用 AWS IAM，您可以安全控制对 AWS 资源的个人访问权限或组访问权限。您可以创建并管理用户身份（“IAM 用户”），并授予这些 IAM 用户访问您资源的 [权限](https://aws.amazon.com/cn/iam/details/manage-permissions/)。您也可以为 AWS 以外的用户（ [联合用户](https://aws.amazon.com/cn/identity/federation/)）授权。
2. What can you do with IAM? (features)
   1. Centralized control of your AWS account
   2. Shared Access to your AWS account
   3. Granular permissions
   4. Identity Federation
      1. Users can log into AWS with same username/password that they log into Active Directory (PC) / Facebook / Linkedin
   5. Multifactor Authentication (MFA)
      1. You need user name & password & special code to log in
   6. Provide temporary access for users/devices and services where necessary
   7. Allows you to set up your own password rotation policy
   8. Integrates with many different AWS services
   9. Supports PCI, DSS compliance
3. Key Terminology for IAM
   1. **Users** - End users (people, employees of an organization)
   2. **Groups** - Collection of users, each user in the group will inherit the permissions of the group
      1. **A group that can access in S3, another group can access EC2**
   3. **Roles** – You create roles and assign then to AWS resources
      1. Allowing one part of AWS (EC2) to do something with another part (S3)
   4. **Policy Documents** – Polices are made up of documents, called Policy documents. These documents are in a format called JSON and they give permissions as to what a User/Group/Role is able to do
      1. JSON format
         1. “\*” means no restrictions, allow you to do anything

### LAB 9: Identity Access Management - IAM

1. IAM is Universal. Does not apply to regions at this time.
2. **Root account -** the account created when first setup your AWS account. It has complete Admin access
3. New users have NO permissions when first created
   1. We need to create an administrator access policy and sign that to the “developer” group
4. New users are assigned and Access Key ID and Secret Access Key when first created
   1. Optimal 🡪 console access or programmatic access
   2. These are not the same as a password
      1. You can’t use the Access key ID & Secret Access Key to login in to the console, but you can use this to access AWS via APIs and Command Line
   3. You only get to view these once
      1. make sure save them (csv) in a secure location
5. Always setup multifactor authentication on root account
   1. If someone hacks your email 🡪 you can do a password reset
      1. 🡪 google authenticator app
6. You can create and customize password rotation policies

### Lab 10: Create A Billing Alarm – CloudWatch

1. My Billing Dashboard
   1. Receive Billing Alerts
2. Cloudwatch
   1. Billing alarm

# Amazon S3

## S3

1. What is S3?
   1. Amazon S3 is object storage built to store and retrieve any amount of data from anywhere on the Internet. It’s a simple storage service that offers an extremely durable, highly available, and infinitely scalable data storage infrastructure at very low costs.
      1. A safe place to store your files.
      2. It is an object-based storage
      3. The data is spread across multiple devices and facilities
2. The Basics of S3
   1. Objected-based 🡪 allows you to upload files
      1. Not suitable to install an operating system/application on your database
   2. files can be from 0 bytes to 5TB.
   3. There is unlimited storage 🡪 extra storage will be paid by gig
   4. Files are stored in “Buckets” 🡪 “folders” in the cloud
   5. S3 is a universal namespace. Each bucket name must be unique globally
      1. https://s3-eu-west-1.amazonaws.com/unique-bucket-name
      2. Remember this format
   6. When you upload a file to S3, you will receive a HTTP 200 code if the upload was successful
3. S3 Object: Key, Value Store
   1. Key - Name of object to be stored
   2. Value - Data inside the file and is made up of a sequence of bytes
   3. Version ID – Important for versioning
   4. Metadata - Data about the data you are storing
   5. Subresource
      1. Access Control Lists
         1. You can put individual permissions on your files
         2. Ex: Only payroll department can see payroll spreadsheet (who can access this object)
      2. Torrents
4. How does data consistency work for S3?
   1. Read after Write consistency for PUTS of new objects.
      1. If you uploaded a file to S3 you are able to read it immediately
   2. Eventual Consistency for overwrite PUTS and DELETES (can take some time to propagate)
      1. Changes to objects can take a little bit of time to propagate
         1. S3 is spread across multiple availability zones
      2. If you update an existing file or delete a file and read it immediately, you may either get the older version or the new version.
5. S3 has the following guarantees from Amazon
   1. Built for 99.99% availability for the S3 platform
   2. Amazon guarantee 99.9% availability
   3. Amazon guarantees 99.99999999999% (11\* 9s) durability for S3 information
      1. XX% chance you won’t lose anything
6. S3 Features
   1. Tiered storage available
      1. Know the different storage classes in S3
   2. Lifecycle management
      1. When a file is over 30 days old, 🡪 we can move it form one storage tier to another 🡪 archive it off to Glacier
   3. Versioning
      1. You can have multiple different versions of a file
   4. Encryption
   5. MFA Delete
      1. Protect objects by turning on multi factor authentication delete
   6. Secure data using Access Control Lists and Bucket Policies
7. S3 - Storage Tiers/Classes
   1. S3 (Standard/Normal) (Frequently accessed)
      1. 99.99% availability
      2. 99.99…%(11\*9’s ) durability
      3. stored redundantly across multiple devices in multiple facilities (AZ)
         1. Same as S3 IA
      4. Designed to sustain the loss of 2 facilities concurrently
   2. **S3 IA (Infrequently Accessed)**
      1. Used for data that is accessed less frequently but still requires rapid access when needed
      2. Lower fee than S3 but charged a retrieval fee
   3. S3 One Zone – IA
      1. Lower-cost option for infrequently access
      2. Do not require the multiple availability zone data resilience
      3. Even lower cost than S3-IA for IA data (cheapest)
   4. S3 – Intelligent Tiering
      1. Designed to optimize cost 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. You can reliably store any amount of data at costs that are competitive with or cheaper than on-premises solutions. Retrieval times 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. S3 Charges
   1. Storage
   2. Requests
   3. Storage Management Pricing
      1. Tag (“only human resources department can own this object”)
   4. Data transfer pricing
   5. Transfer Acceleration
   6. Cross Region Replication Pricing
9. S3 Transfer Acceleration
   1. Enables fast, easy and secure transfers of files over long distances between your end users and an S3 bucket.
   2. It takes advantage of AWS CloudFront globally distributed edge locations.
      1. Instead of upload files directly to the S3 bucket, they actually upload it to closest an edge location. 🡪 data is routed to Amazon S3 over an optimized network path.

### Lab 12: Create an S3 Bucket

1. Buckets are a universal name space
2. Upload an object to S3 receive a HTTP 200 Code
3. S3, S3-IA, S3-IA (One Zone), Glacier
4. Control access to buckets using either a bucket ACL or using Bucket Polices

## Security & Encryption

1. Securing your buckets
   1. By default, all newly created buckets are **PRIVATE**.
   2. You can setup access control to your buckets using:
      1. **Bucket Policies**
      2. **Access Control Lists**
   3. S3 buckets can be configured to create access logs which log all requests made to the S3 bucket. This can be done to another bucket through another AWS account
2. Types of Encryption
   1. **In Transit** – when you send information to and from your bucket
      1. Using HTTPS 🡪 all the traffic/files that you’re uploading are be encrypted in transit
      2. Achieved by SSL/TLS.
   2. **At Rest – when we encrypt that data that’s being stored**
      1. Server Side – Amazon help you encrypt the object
         1. **SSE-S3** - S3 Managed keys.
            1. **S3 handles all our encryption for us**
         2. **SSE KMS** - AWS Key Management Service, Managed Keys.
            1. **Using keys from KMS**
         3. **SSE-C** - Server-Side Encryption with Customer Provided Keys.
            1. **You provide the keys and you manage the encryption**
      2. Client Side – where you encrypt the object and upload it to S3

### Lab 14: Version Control

1. Stores all versions of an object (including all writes and even if you delete an object)
2. Great backup tool
3. Once enabled, versioning can’t be disabled, only suspended
4. Integrates with Lifecycle rules
5. Version’s MFA Delete capability, which uses multi-factor authentication, can be used to provide an additional layer of security

### Lab 15: Lifecycle Management and Glacier

1. Automates moving your objects between the different storage tiers
2. Can be used in conjunction with versioning
3. Can be applied to current versions and previous versions
4. Example
   1. Transition to the Standard – Infrequent Access Storage Class (30 days after the creation date)
   2. Archive to the Glacier Storage Class (30 days after IA, if relevant)
   3. Permanently Delete

### Lab 16: Cross Region Replication

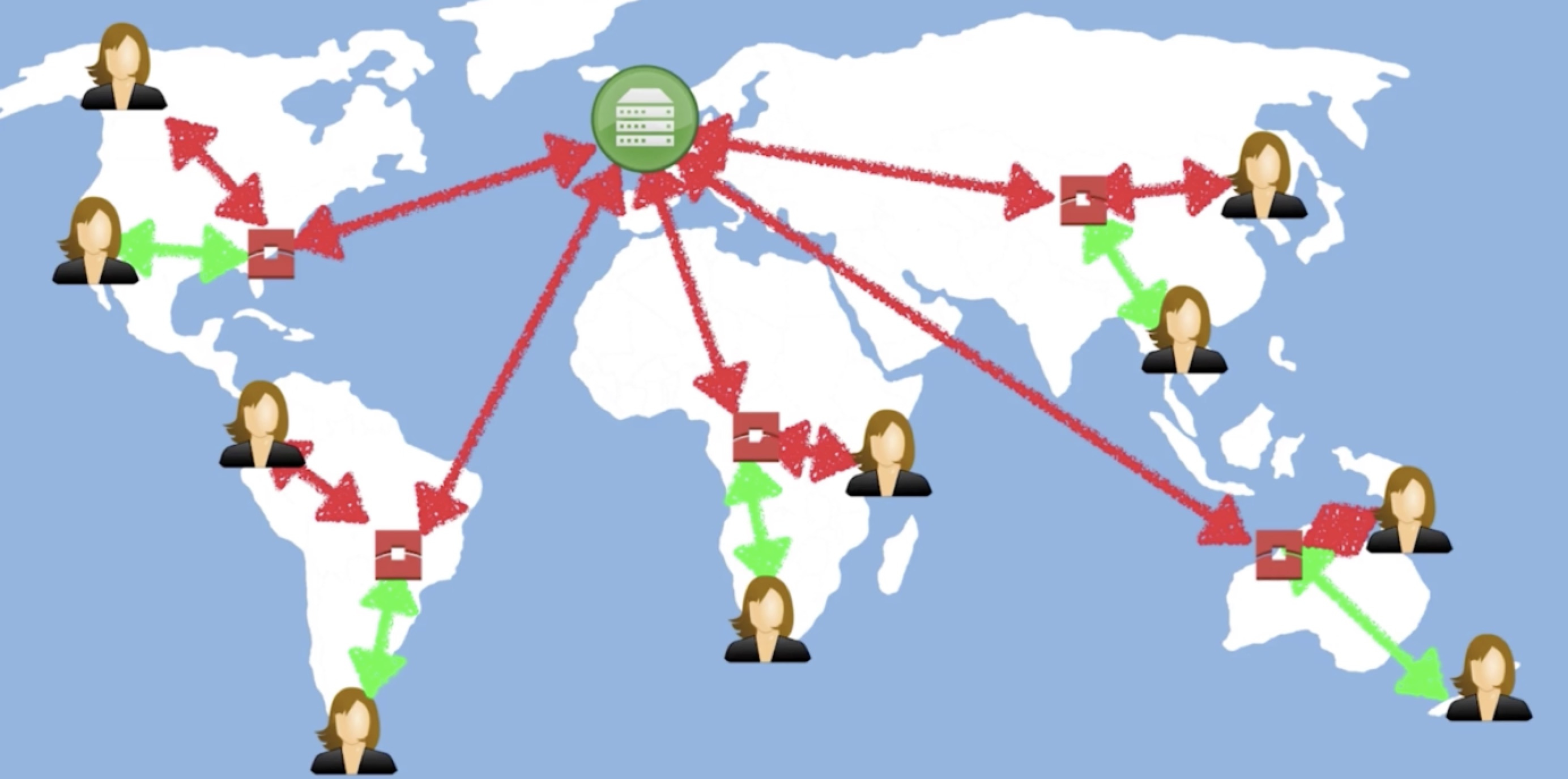
1. Versioning must be enabled on both the source and destination buckets
2. Regions must be unique
3. Files in an existing bucket are not replicated automatically.
4. All subsequent updated files will be replicated automatically
5. Delete markers are NOT replicated
6. Deleting individual versions or delete markers will not be replicated

## S3 Transfer Acceleration

1. S3 transfer acceleration utilities the CloudFront Edge Network to accelerate your uploads to S3. Instead of uploading directly to your S3 bucket, you can use a distinct URL to upload directly to an edge location which will then transfer that file to S3. You will get a distinct URL to upload to
   1. acloudguru.s3-accelerate.amazonaws.com
2. Properties 🡪 Enable Transfer acceleration

## CloudFront CDN

1. What is CDN?
   1. A content delivery network (CDN) is a system of distributed servers (network) that deliver webpages and other web content to a user based on the geographic locations of the user, the origin of the webpage, and a content delivery server
2. CDN - Key Terminology
   1. **Edge Location** - Location where content will be cached. Separate to an AWS region/Availability Zone (cached 🡪 “save”)
      1. Edge Locations are not just for READ only 🡪 you can write/put an object on them
      2. Object are cached for life of TTL (Time To Live)
      3. You can clear cached objects, but you will be charged
   2. **Origin** - This is the origin of all the files that the CDN will distribute. Can be S3 bucket, EC2 instance, Elastic Load Balancer or Route 53.
   3. **Distribution** - The name given the CDN which consists of a collection of Edge Locations
      1. **Web Distribution** - Typically used for websites
      2. **RTMP (Real Time Message Protocol)** - Used for media streaming
3. What is CloudFront
   1. CloudFront can be used to deliver your entire website, including dynamic content, static, streaming and interactive content using a global network of edge locations.
   2. Requests for your content are automatically routed to the nearest Edge Location, so content is delivered with the best possible performance.
   3. CloudFront is optimized to work with other Amazon Web Services like S3, EC2, Elastic Load Balancing and Route 53. CloudFront also works seamlessly with any non-AWS origin server which stores the original, definitive versions of your files.



### Lab 18: Create a CloudFront Distribution

1. Category: Networking & Content Delivery
2. Create CloudFront
   1. Can modify TTL
   2. You can restrict viewer access
3. Done 🡪 Copy CDN’s domain name + “/” + file name (in S3) to browser

## Snowballs

1. **Snowball**
   1. Petabyte-scale data transport solution that uses secure appliances to transfer large amounts of data into and out of AWS.
   2. Using Snowball addresses common challenges with large-scale data transfers including high network costs, long transfer times, and security concerns.
   3. Transferring data with Snowball is simple, fast, secure and can be as little as 1/5 the cost of high-speed internet.
   4. Snowball comes in either a 50TB or 80TB size. Snowball uses multiple layers of security designed to protect your data including tamper-resistant enclosures, 256-bit encryption, and an industry-standard Trusted Platform Module (TPM) designed to ensure both security and full chain-of-custody of your data.
   5. Once the data transfer job has been processed and verified, AWS performs a software erasure of the Snowball appliance
   6. Snowball can import to S3 and also export from S3
2. **Snowball Edge**
   1. Snowball Edge is a 100TB data transfer device with on-board storage and compute capabilities. You can use Snowball Edge to move large amounts of data into and out of AWS, as a temporary storage tier for large datasets, or to support local workloads in remote or offline locations.
   2. Snowball Edge connects to your existing applications and infrastructure using standard storage interfaces, streamlining the data transfer process and minimizing setup and integration.
   3. Snowball Edge can cluster together to form a local storage tier and process your data on-premises, helping ensure your applications continue to run even when they are not able to access the cloud.
3. **Snowmobile**
   1. Snowmobile is an Exabyte-scale data transfer service used to move EXTREMELY large amounts of data to AWS.
   2. You can transfer up to 100PB per Snowmobile, a 45ft long ruggedized shipping container, pulled by a semi-truck.
   3. Snowmobile makes it easy to move massive volumes of data to the cloud, including video libraries, image repositories, or even a complete data center migration. Transferring data with Snowmobile is secure, fast and cost effective.

### Lab 21: Snowball

1. Category: Migration & Transfer

## Storage Gateway

1. What is Storage Gateway?
   1. A Service that connects an on-premise software appliance with cloud-based storage to provide seamless and secure integration between an organization's on-premise IT environment and AWS's storage infrastructure. The service enables you to securely store data to AWS cloud for scalable and cost-effective storage.
   2. The AWS storage Gateway’s software appliance is available for downloaded as virtual machine (VM) image that you install on a host in your datacenter. Storage Gateway supports either VMware ESXi or Microsoft Hyper-V. Once you've installed your gateway and associate with AWS account through the activation process, you can use the AWS Management Console to create the storage gateway option that is right for you.
2. Three Types of Gateway Storage
   1. File Gateway (NFS) - For flat files, stored directly to S3 (min cost)
      1. Store flat files in S3 accessed through a Network File System (NFS) mount point.
      2. Ownership, permissions, and timestamps are durably stored in S3 in the user-metadata of the object associated with the file.
      3. Once objects are transferred to S3, they can be managed as native S3 objects, and bucket policies such as versioning, lifecycle management, and cross-region replication apply directly to objects stored in your bucket.
   2. Volumes Gateway (iSCSI)
      1. The volume interface presents your applications with disk volumes using the iSCSI block protocol.
         1. (Virtual) hard disk
      2. Data written to these volumes can be asynchronously backed up as point-in-time snapshots of your volumes and stored in the cloud as AWS EBS (Elastic Block Store - VM) snapshots.
      3. Snapshots are incremental backups that capture only the changed blocks.
      4. All snapshot storage is also compressed to minimize your storage charges.
      5. Type one - Stored Volumes
         1. Stored volumes let you store your primary data locally, while asynchronously backing up that data to AWS.
            1. Complete copy of entire data
         2. Stored volumes provide your on-premise applications with low-latency access to their entire datasets, while providing durable, off-site backups.
         3. You can create storage volumes and mount them as iSCSI devices from your on-premises application servers.
         4. Data written to your stored volumes is stored on your on-premises storage hardware.
         5. Entire dataset is stored on site and is asynchronously backed up to S3 in the form of AWS EBS (Elastic Block Store) snapshots 1 GB - 16 TB in size for Stored Volumes.
      6. Type Two - Cached Volumes
         1. Cached volumes let you use S3 as your primary data storage while retaining frequently accessed data locally in your storage gateway.
            1. Entire dataset is stored on S3 and the most frequently accessed data is cached on site.
         2. Cached volumes minimize the need to scale your on-premise storage infrastructure, while still providing your applications with low-latency access to their frequently accessed data.
         3. You can create storage volumes up to 32Tb in size and attach to them as iSCSI devices from your on-premises application servers. Your gateway stores data that you write to these volumes in S3 and retains recently read data in your on-premises storage gateways cache and upload buffer storage. 1 GB - 32 TB size for cached volumes.
   3. Tape Gateway – used be called “Gateway Virtual Type Library (VTL)”
      1. Offers a durable, cost-effective solution to archive your data in AWS cloud. The VTL interface it provides lets you leverage your existing tape-based backup application infrastructure to store data on virtual tape cartridges that you create on your tape gateway.
      2. Each tape gateway is preconfigured with a media changer and tape drivers, which are available to your existing client backup applications as iSCSI devices. You add tape cartridges as you need to archive your data. Supported by Netbackup, Backup Exec, Veeam etc.

# Ec2 – Elastic compute cloud

## EC2

1. What is EC2 (Elastic Cloud Compute)?
   1. AWS EC2 is a web service that provides resizable compute capacity (virtual machine) in the cloud. EC2 reduces the time required to obtain and boot new server instances to minutes, allowing you to quickly scale capacity, both up and down, as your computing requirements change.
   2. EC2 has changed the economics of cloud computing by allowing you to pay only for capacity that your actually use. EC2 provides developers the tools to build failure resilient applications and isolate themselves from common failure scenarios.
2. EC2 Pricing Options
   1. On Demand - Allows you to pay a fixed rate by the hour (or by the second) with no commitment.
      1. **Use Cases:**
      2. Perfect for users that want the low cost and flexibility of EC2 without any of the up-front payment or long-term commitment
      3. Applications with short term, spikey or unpredictable workloads that cannot be interrupted
      4. Applications being developed or tested on EC2 for the first time
   2. Reserved - Provides you with a capacity reservation and offer a significant discount on the hourly charge for an instance. 1 year or 3 year terms.
      1. **Use Cases**
         1. Applications with steady state or predictable usage
         2. Applications that require reserved capacity
         3. Users can make up front payments to reduce their total computing costs even further
      2. Three pricing types
         1. Standard Reserved Instanced (RIs)
            1. Up to 75% off on-demand instances
            2. The more you pay up front and the longer the contract, the greater the discount
         2. Convertible Reserved Instanced
            1. Up to 54% off on demand capability to change the attributes of the RI as long as the exchange results in the creation of Reserved Instances of equal or greater value.
            2. Allow to change between EC2 instance types (see below)
         3. Scheduled Reserved Instanced
            1. Available to launch within the time window you reserve. This option allows you to match your capacity reservation to predictable recurring schedule that only requires a fraction of a day, a week, or a month.
   3. Spot - Enables you to bid whatever price you want for an instance capacity, providing for even greater savings if your applications have flexible start and end times.
      1. **Use Cases:**
         1. Applications that have flexible start and end times
         2. Applications that are only feasible at very low compute prices
         3. Users with an urgent need for a large amount of additional computing capacity.
      2. Important: If a Spot instance is terminated by Amazon EC2, you will not be charged for a partial hour of usage. However, if you terminate the instance yourself, you will be charged for the complete hour in which the instance ran. (Important)
   4. Dedicated Hosts - Physical EC2 server dedicated for your use. Dedicated Hosts can help you reduce costs by allowing you to use your existing server-bound software licenses.
      1. **Use Cases**:
         1. Useful for regulatory requirements that may not support multitenant virtualization.
         2. Great for licensing which does not support multi-tenancy or cloud deployments
         3. Can be purchased On-Demand (hourly).
         4. Can be purchased as a Reservation for up to 70% off the On-Demand price.
3. EC2 Instance Types
   1. **No need to memorize for associate exams**

| **Family** | **Specialty** | **Use Cases** |
| --- | --- | --- |
| F1 | Field Programmable Gate Array | Genomics research, financial analytics, real-time video processing, big data etc |
| I3 | High Speed Storage | NoSQL DBs, Datawarehousing |
| G3 | Graphics Intensive | Video Encoding / 3D Application Streaming |
| H1 | High Disk Throughput | MapReduce-based workloads, distributed file systems such as HDFS and MapR-FS |
| T3 | Lowest Cost General Purpose | Web Servers / Small DBs |
| D2 | Dense Storage | Fileservers / Data Warehousing / Hadoop |
| R5 | Memory Optimization | Memory Intensive Apps/DBs |
| M5 | General Purpose | Application Servers |
| C5 | Compute Optimized | CPU Intensive Apps / DBs |
| P3 | Graphics / General Purpose GPU | Machine Learning, Bit Coin Mining etc |
| X1 | Memory Optimized | SAP HANA / Apache Spark |
| Z1D | High compute capacity and a high memory footprint | Ideal for electronic design automation (EDA) and certain relational database workloads with high per-core licensing costs |
| A1 | Arm-based workloads | Scale-out workloads such as web servers |
| U-6tb1 | Bare Metal | Bare metal capabilities that eliminate virtualization overhead |

### Lab 25: Launch an EC2 Instance

1. Category: Compute
2. Termination protection is turned off by default, you **MUST** turn it on.
3. On an EBS-backed instance, the default action is for the root EBS volume to be deleted when the instance is terminated
4. EBS Root Volume of you DEFAULT AMI's cannot be encrypted. You can also use a third-party tool (such as bit locker) to encrypt the root volume, or this can be done when creating AMI's (future lab) in the AWS console or using the API.
5. Additional volumes can be encrypted.

### Lab 27: Security Groups

1. What is a Security Group?
   1. A security group is a virtual firewall that's controlling traffic to your EC2 instance. When you first launch as EC2 instance you associate it to 1 or more instances. You have the ability to add rules to these security groups that allows traffic to or from these instances.
2. All inbound traffic is blocked by default
3. All outbound traffic is allowed
4. Changes to security group take effect immediately
5. You can have any number of EC2 instances within a security group.
6. You can have multiple security groups attached to EC2 instances.
7. Security groups are STATEFUL.
8. If you create an inbound rule allowing traffic in, that traffic is automatically allowed back out again
9. You cannot block specific IP addresses using Security Groups, use Network Access Control Lists.
10. You can specify allow rules, but not deny rules.

## EBS

1. What is EBS (Elastic Block Storage)?
   1. Amazon EBS provides persistent block storage volumes for use with Amazon EC2 instance in the AWS Cloud.
   2. Each Amazon EBS volume is automatically replicated within its Availability Zone to protect you from component failure, offering high availability and durability
   3. EC2: Virtual server in cloud, EBS: Virtual hard disk
      1. A disk in the cloud that you attach to your EC2 instances, like a C/D/E drive in a computer
2. 5 different EBS Volume Types
   1. General Purpose SSD (API Name: GP2)
      1. General purpose that balances price and performance for a wide variety of transactional workloads
      2. Case: Most work loads
      3. Max IOPS volume 🡪 16,000
   2. Provisioned IOPS SSD (IO1)
      1. Highest-performance SSD volume designed for mission-critical applications
      2. Case: Databases
      3. Max IOP volume 🡪 64,000
   3. Throughput Optimized Hard Disk Drive (HDD) (ST1)
      1. Low cost HDD volume designed for frequently accessed throughout intensive workloads
      2. Case: Big Data & Data warehouses
      3. Max IOP volume 🡪 500
   4. Cold HDD (SC1)
      1. Lowest cost HDD volume designed for less frequently accessed workloads
      2. Use Case: File server
      3. Max IOP volume 🡪 250
   5. Magnetic (Standard)
      1. Previous generation HDD
      2. Case: Workloads where data is infrequently accessed
      3. Max IOP volume 🡪 40-200

### Lab 29: EBS Volumes & Snapshots

1. EBS Snapshots
   1. Volumes exist on EBS.
      1. EBS: is a Virtual Hard Disk
   2. Snapshots exist on S3
      1. Snapshots: a photograph of the disk
   3. Snapshots are a point in time copies of Volumes
   4. Snapshots are incremental - this means that only the blocks that have changed since your last snapshot are moved to S3.
   5. If it's your first snapshot, it may take time to create
   6. To create a snapshot of Amazon EBS volumes that serve as root devices, you should stop the instance before taking the snapshot
   7. However, you can take a snapshot while the instance is running.
   8. You can create AMI's from both Volumes and Snapshots.
      1. AMI – Amazon Machine Image
   9. You can change EBS volume sizes on the fly, including changing the size and storage type.
   10. Volumes will **ALWAYS** be in the same availability zone as the EC2 instance.
2. Migrating EBS
   1. To move and EC2 volume from one AZ/Region to another, take a snap or an image of it, create an AMI from the snapshot and then use the AMI to launch the EC2 instance in a new AZ/Region.
   2. To move an EC2 volume from one region to another, take a snapshot of it, create an AMI from the snapshot and then copy the AMI from one region to the other. Then use the copied AMI to launch the new EC2 instance in the new region

## AMI Types (EBS vs Instance Store)

1. What should you select your AMI based on?
   1. Region and availability zones
   2. Operating system
   3. Architecture (32/64-bit)
   4. Launch Permissions
   5. Storage for the Root Device (Root Device Volume)
      1. EBS Backed Volumes
      2. Instance Store (Ephemeral Store)
2. EBS vs. Instance Store
   1. All AMIs are categorized as either backed by Amazon EBS or backed by instance store.
   2. **For EBS Volumes:**
      1. The root device for an instance launched from the AMI is an Amazon EBS volume created from an Amazon EBS snapshot.
   3. **For Instance Store Volumes:**
      1. The root device for an instance launched from the AMI is an instance store volume created from a template stored in Amazon S3.
3. Exam Tips
   1. Instance store volumes are sometimes called *Ephemeral Storage*.
   2. Instance store volumes cannot be stopped. If the underlying host fails, you will lose all your data.
   3. EBS backed instances can be stopped. You will not lose the data on this instance if it is stopped.
   4. You can reboot both, you will not lose your data.
   5. By default, both ROOT volumes will be deleted on termination, however with EBS volumes, you can tell AWS to keep the root device volume.

### Lab 31: Encrypted Root Device Volumes\_Snapshots

1. Snapshots of encrypted volumes are encrypted automatically.
2. Volumes restored from encrypted snapshots are encrypted automatically.
   1. Once you encrypt a snapshot, it’s always going to remain encrypted
3. You can share snapshots, but only if they are unencrypted.
4. These snapshots can be shared with other AWS accounts or made public.
5. Steps to encrypt the root device
   1. Create a Snapshot of the unencrypted root device volume
   2. Create a copy of the Snapshot and select the encrypt option
   3. Create an AMI from the encrypted Snapshot
   4. Use that AMI to launch new encrypted instances

## Cloudwatch

1. What is cloud watch?
   1. CloudWatch is used for monitoring performance
   2. CloudWatch can monitor most of AWS as well as applications that run on AWS
   3. CloudWatch with EC2 will monitor events every 5 minutes by default
   4. You can have 1-minute intervals by turning on detailed monitoring
   5. You can create CloudWatch alarms which trigger notifications
2. What can CloudWatch monitor?
   1. Compute
      1. EC2 Instances
      2. Autoscaling Groups
      3. Elastic Load Balancers
      4. Route53 Health Checks
   2. Storage & Content Delivery
      1. EBS Volumes
      2. Storage Gateways
      3. CloudFront
   3. Host Level
      1. CPU
      2. Network
      3. Disk
      4. Status Check
3. What is AWS Cloud Trail?
   1. AWS CloudTrail increases visibility into your user and resource activity by recording AWS Management Console actions and API calls. You can identify which users and accounts called AWS, the source IP address from which the calls were made, and when the calls occurred.
      1. CloudWatch monitors performance
      2. CloudTrail monitors API calls in the AWS platform

### Lab 33: CloudWatch

1. Create Instance 🡪 Configure Instance 🡪 monitoring 🡪 “Enable CloudWatch detailed monitoring”
2. Standard Monitoring = 5 minutes
3. Detailed Monitoring = 1 minute
4. What can you do with CloudWatch?
   1. **Dashboards** - Creates awesome dashboards to see/monitor what is happening with your AWS environment.
   2. **Alarms** - Allows you to set Alarms that notify you when a particular threshold are hit.
   3. **Events** - Helps you to respond to state changes in your AWS resources.
   4. **Logs** - Helps you to aggregate, monitor and store logs.
5. Don’t confuse with cloud trial
   1. CloudWatch – monitoring all the services performance within AWS
   2. CloudTrial – auditing what people are doing with AWS account

### Lab 34: AWS Command Line (CLI)

1. You can interact with AWS from anywhere in the world just by using the command line (CLI)
2. You will need to set up access in IAM

## Identity Access Management Roles

1. Roles are more secure than storing your access key and secret access key on individual EC2 instances
2. Roles are easier to manage
3. Roles can be assigned to EC2 instance after it is created using both the console & command line
4. Roles are universal 🡪 you can use them in any region

### Lab 36: Using Boot Strap Scripts

1. Bootstrap scripts run when an EC2 instance first boots
2. Can be a powerful way of automating software installs and updates

### Lab 37: EC2 Instance Metadata

1. Metadata is used to get information about an instance (such as public ip)

### Lab 38: EFS (Elastic File System)

1. What is EFS?
   1. AWS EFS is file storage service for AWS EC2 instances. Amazon EFS is easy to use and provides a simple interface that allows you to create and configure file systems quickly and easily. With AWS EFS, storage capacity is elastic, growing and shrinking automatically as you add and remove files, so your applications have the storage they need, when they need it.
2. Category: Storage
3. EFS Features
   1. Supports the Network File System version 4 (NFSv4) protocol
   2. You only pay for the storage you use (no pre-provisioning required)
   3. Can scale up to the petabytes
   4. Can support thousands of concurrent NFS connections
   5. Data is stored across multiple AZ's within a region
   6. Read After Write Consistency

## EC2 Placement Groups

1. What is a Placement Group?
   1. **Clustered Placement Group**
      1. A cluster placement group is a grouping of instances within a **single** Availability Zone.
      2. Placement groups are recommended for applications that need low network latency, high network throughput, or both.
      3. Only certain instances can be launched in to a Clustered Placement Group
   2. **Spread Placement Group**
      1. Opposite of a Clustered Placement Group. A Spread Placement Group is a group of instances that are each placed on distinct underlying hardware.
      2. Spread Placement Groups are recommended for applications that have a small number of critical instances that should be kept separate from each other.
2. Exam Tips
   1. A clustered placement group can’t span multiple Availability Zones
   2. A spread placement group can
   3. The name you specify for a placement group must be unique within your AWS account
   4. Only certain types of instances can be launched in a placement group (Compute Optimized, GPU, Memory Optimized, Storage Optimized)
   5. AWS recommend homogenous instances within placement groups
   6. You can’t merge placement groups
   7. You can’t move an existing instance into a placement group. You can create an AMI from your existing instance, and then launch a new instance from the AMI into a placement group

# Databases

## Databases

1. What are a Relational Databases (RDS)?
   1. Relational databases are what most of us are all used to. They have been around since the 70's and you can think about them like spreadsheets:
      1. Database (excel)
      2. Tables (worksheet)
      3. Columns/Fields
      4. Rows
   2. Relational Databases Examples
      1. SQL Server
      2. Oracle
      3. MySQL
      4. PostgreSQL
      5. Aurora
      6. MariaDB
   3. Two key features:
      1. Multi-AZ 🡪 For Disaster Recovery
      2. Read Replicas 🡪 For Performance
2. Non-Relational Databases (DynamoDB (NoSQL))
   1. Difference – You can have NA, instead of keep consistency for records
   2. Structure
      1. Table => Collection
      2. Row => Document
      3. Columns => Key, Value Pairs
   3. **NON-RELATIONAL Databases Examples**

{

"\_id": "394ejojaj903091881dnna",

"name": "nigel",

"age": 30,

"address": [

{"Street":"21 Jump Street",

"suburb":"Richmond"}

]

}

* 1. Example: DynamoDB (No SQL)

1. Data Warehousing
   1. Used for business intelligence. Tools like Cognos, Jaspersoft, SQL Server, Reporting Services, Oracle Hyperion, SAP NetWeaver.
   2. Used to pull in very large and complex data sets. Usually used by management to do queries on data (such as current performance vs targets etc).
2. OLTP (Online Transaction Processing) vs. OLAP (Online Analytics Processing)
   1. OLTP Example
      1. Order number: 2120121
      2. Pulls up a row of data such as Name, Date, Address to Deliver to, Delivery Status etc.
   2. OLAP Example
      1. Net Profit of given product or device
      2. Pulls in large number of records
         1. Sum of products sold in region
         2. Sum of products sold in continent
         3. Unit cost of product in each region
         4. Sales price of each product
         5. Difference between sales price and unit cost
   3. Data Warehousing databases use different type of architecture both from a database perspective and infrastructure layer.
      1. Amazon’s Data Warehouse Solution is called Redshift
      2. Redshift for Business Intelligence or Data Warehousing
3. Elasticache
   1. ElastiCache is a web service that makes it easy to deploy, operate and scale an in-memory cache in the cloud. The service improves the performance of web applications by allowing you to retrieve information from fast, managed, in-memory caches, instead of relying entirely on slower disk-based databases.
   2. Elasticache to speed up performance of existing databases (frequent identical queries)
      1. Example: When people visit Amazon.com, they might have a particular store with top 10 purchased items. If thousand people are visiting 🡪 the top 10 is not really going to that much 🡪 New customer can directly give the information from Elasticache 🡪 takes a high load off of your databases
   3. ElasticCache supports two open-source in-memory caching engines.
      1. Memcached
      2. Redis

### Lab 42: Create RDS Instance

1. Category: Database 🡪 RDS
2. RDS runs on virtual machines
3. You can’t log in to these operating systems however
4. Patching of the RDS Operating System and DB is Amazon’s responsibility
5. RDS is NOT Serverless
6. Aurora Serverless IS Serverless

## RDS – Back Ups, Multi-AZ & Read Replicas

1. There are two types of Buckups for RDS:
   1. Automated Backups
      1. Automated Backups allow you to recover your database to any point in time within a 'retention period'. The retention period can be between one and 35 days.
      2. Automated Backups will take a full daily snapshot and will also store transaction logs throughout the day.
      3. When you do a recovery, AWS will first choose the most recent daily backup, and then apply transaction logs relevant to that day. This allows you to do a point in time recovery down to a second, within a retention period.
      4. Automated Backups are enabled by default. The backup data is stored in S3 and you get free storage space equal to the size of your database. So if you have an RDS instance of 10Gb, you will get 10Gb worth of storage.
      5. Backups are taken within a defined window. During the backup window, storage I/O may be suspended while your data is being backed up and you may experience elevated latency
   2. Database Snapshots
      1. DB Snapshots are done manually (ie they are user initiated) They are stored even after you delete the original RDS instance, unlike automated backups.
         1. When you delete your RDS instance, it will ask you if you want to take a final snapshot before deleting
2. Restoring Backups
   1. Whenever you restore either an Automatic Backup or a manual Snapshot, the restored version of the database will be a new RDS instance with a new DNS endpoint
      1. original.us-west-1.rds.amazonaws.com -> restored.eu-west-1.rds.amazonaws.com
3. Encryption
   1. Encryption at rest is supported for MySQL, Oracle, SQL Server, PostgreSQL, MariaDB & Aurora.
   2. Encryption is done using the AWS Key Management System (KMS) service. Once your RDS instance is encrypted, the data stored at rest in the underlying storage is encrypted, as are its automated backups, read replicas and snapshots.
4. What is Multi-AZ
   1. Multi-AZ allows you to have an exact copy of your production database in another Availability Zone. AWS handles the replication for you, so when your production database is written to, this write will automatically be synchronized to the stand by database.
   2. In the event of planned database maintenance, DB instance failure, or AZ failure, RDS will automatically failover to the standby so that database operations can resume quickly without admin intervention.
   3. **NOTE:** Multi-AZ is for disaster recovery only. It is not primarily used for improving performance. For performance improvement, you need **Read Replicas**
   4. Available for the following databases:
      1. SQL Server
      2. Oracle
      3. MySQL Server
      4. PostgreSQL
      5. MariaDB
5. Read Replicas
   1. Read replicas allow you to have a read-only copy of your production database. This is achieved by using asynchronous replication from the primary RDS instance to the Read Replica. You use Read Replicas primarily for very read-heavy database workloads.
   2. Available for the following databases:
      1. MySQL Server
      2. PostgreSQL
      3. MariaDB
      4. Aurora
   3. Things to know about Read Replicas:
      1. Used for scaling, not disaster control!
      2. Must have automatic backups turned on in order to deploy a Read Replica
      3. You can have up to 5 Read Replica copies of any database.
      4. You can have Read Replicas of Read Replicas (inception) – watch out for latency
      5. Each Read Replica will have its own DNS end point.
      6. You can have Read Replicas that have Multi-AZ
      7. You can create Read Replicas of Multi-AZ source databases
      8. Read Replicas can be promoted to be their own databases. This breaks the replication.
      9. You can have a Read Replica in a second region.

### Lab 44: RDS – Backups, Multi-AZ & Read Replicas

1. Category: Databases 🡪 RDS
2. Two types of Backups for RDS
   1. Automated Backups
   2. Database Snapshots
3. Read Replicas
   1. Can be Multi-AZ
   2. Used to increase performance
   3. Must have backups turned on
   4. Can be in different regions
   5. Can be Aurora or MySQL
   6. Can be promoted to master, this will break the Read Replica
4. MultiAZ
   1. Used for disaster recovery
   2. You can force a failover from one AZ to another by rebooting the RDS instance
5. Encryption

## DynamoDB

1. Amazon DynamoDB is a fast and flexible NoSQL database service for all applications that need consistent, single-digit millisecond latency at any scale. It is a fully managed database and supports both document and key-value data models. Its flexible data model and reliable performance make it a great fit for mobile, web, gaming, ad-tech, IoT and many other applications.
2. The basics of DynamoDB:
   1. Stored on SSD Storage
   2. Spread across **3** geographically distinct data centers
3. Two different types of read models
   1. Eventual Consistent Read (Default)
      1. Consistency across all copies of data is usually reached within seconds. Repeating a read after a short amount of time should return the updated data. (Best Read Performance)
   2. Strongly Consistent Read
      1. A strongly consistent read returns a result that reflects all writes that received a successful response prior to the read.
      2. Need to read in one second or less

## Redshift

1. What is Redshift?
   1. Amazon Redshift is a fast and powerful, fully managed petabyte-scale data warehouse service in the cloud.
   2. Customers can start small for just $0.25 per hour with no commitments or upfront costs and scale to a petabyte or more for $1,000 per terabyte per year, less than 1/10 of most data warehousing solutions.
   3. Redshift is used for business intelligence
2. OLAP transaction Example:
   1. Net Profit of given product or device
   2. Pulls in large number of records
      1. Sum of products sold in region
      2. Sum of products sold in continent
      3. Unit cost of product in each region
      4. Sales price of each product
      5. Sales price - unit cost
3. Data Warehousing databases use different type of architecture both from a database perspective and infrastructure layer
   1. Amazon’s Data Warehouse Solution Is Called Redshift
4. Redshift can be configured as follows:
   1. Single Node (160Gb)
   2. Multi-Node
      1. Leader Node (manages client connections and receives queries)
      2. Compute Node (store data and perform queries and computations) - Up to 128 Compute Nodes
5. **Advanced Compression:**
   1. Columnar data stores can be compressed much more than row-based data stores because similar data is stored sequentially on disk.
      1. compressed individual columns
   2. Redshift employs multiple compression techniques and can often achieve significant compression relative to traditional relational data stores. In addition, Redshift doesn't require indexes or materialized views and so uses less space than traditional relational database systems.
   3. When loading data into an empty table, Redshift automatically samples your data and selects the most appropriate compression scheme.
6. **Massive Parallel Processing (MPP)**
   1. Redshift automatically distributes data and query load across all nodes. Redshift makes it easy to add nodes to your data warehouse and enables you to maintain fast query performance as your data warehouse grows.
7. Redshift Backups
   1. Enabled by default with a 1 day retention period
   2. Maximum retention period is 35 days
   3. Redshift always attempts to maintain at least three copies of your data (the original and replica on the compute nodes and a backup in Amazon S3)
   4. Redshift can also asynchronously replicate your snapshots to S3 in another region for disaster recovery.
8. Redshift is priced as follow:
   1. Compute Node Hours
      1. Total number of hours you run across all your compute nodes for the billing period
      2. Billed for 1 unit per node per hour, so a 3 - node data warehouse cluster running persistently for an entire month would incur 2,160 instance hours.
      3. You will **not** be charged for leader node hours; only compute nodes will incur charges
   2. Backups
   3. Data transfers (Only within a VPC, not outside of it)
9. Security Considerations:
   1. Encrypted in transit using SSL
   2. Encrypted at rest using AES-256 encryption
   3. By default, Redshift takes care of key management
      1. Manages your keys through HSM (Hardware Security Module)
      2. AWS Key Management Service (KMS)
10. Redshift Availability:
    1. Currently only available in 1 AZ
    2. Can restore snapshots to new AZ's in the event of outage.

## Aurora

1. What is Aurora?
   1. Aurora is a MySQL-compatible, relational database engine that combines the speed and availability of high-end commercial databases with the simplicity and cost effectiveness of open source databases. Aurora provides up to 5x better performance than MySQL at a price point of 1/10 that of a commercial database while delivering similar performance and availability
2. Things to know about Aurora
   1. Start with 10G, Scales in 10G increments to 64 TB (Storage Autoscaling)
   2. Compute resource can scale up to 32vCPUs and 244G of Memory.
   3. Always maintains two copies of your data are contained in each availability zone, with minimum of 3 AZ 🡪 6 copies of your data!
3. Scaling Aurora
   1. Designed to transparently handle the loss of up to 2 copies of data without affecting database write availability and up to 3 copies without affecting read availability.
   2. Aurora storage is also self-healing. Data blocks and disks are continuously scanned for errors and repaired automatically.
4. Two types of Replicas are available:
   1. Aurora Replicas – currently 15
      1. Automated failover is only available with Aurora Replicas
   2. MySQL Replicas – currently 5
5. Backups with Aurora
   1. Automated backups are always enabled on Amazon Aurora DB instances (default). Backups do not impact database performance
   2. You can also take snapshots with Aurora. This also does not impact on performance
   3. You can share Aurora Snapshots with other AWS accounts

## Elasticache

1. ElastiCache is a web service that makes it easy to deploy, operate and scale an in-memory cache in the cloud. The service improves the performance of web applications by allowing you to retrieve information from fast, managed, in-memory caches, instead of relying entirely on slower disk-based databases.
   1. Use Elasticache to increase datebase and web application performance
2. ElastiCache supports two open-source in-memory caching engines:
   1. Both
      1. Simple Cache to offload DB
   2. Memcached (simple)
      1. Ability to scale horizontally
      2. Multi-threaded performance
   3. Redis (advanced)
      1. Advanced data types
      2. Ranking/Sorting datasets
      3. Pub/Sub capabilities
      4. Peresistence
      5. Multi-AZ
      6. Backup & Restore Capabilities

# Route 53

## DNS

1. What is DNS? (Domain Name Service)
   1. If you've used the internet, you've used DNS. DNS is used to convert human friendly domain names (http://acloud.guru) into an Internet Protocol (IP) address (http://92.123.92.1)
   2. IP addresses are used by computers to identify each other on the network. IP addresses commonly come in 2 different forms, **IPv4** and **IPv6**
2. IPv4 vs IPv6
   1. The IPv4 space is 32 bit field and has over 4 billion different addresses (4,294,967,296)
      1. IPv4 Addresses are running out…
   2. IPv6 was created to solve this the depletion issue and has an address space of 128 bits - which is in theory **340,282,366,920,938,463,463,374,607,431,768,211,456** different addresses! Or 340 undecillion addresses
   3. Elastic Load Balance's do not have pre-defined IPv4 addresses, you resolve to them using a DNS name
3. Top Level Domains
   1. If we look at common domain names such as google.com, bbc.co.uk.acloud.guru etc. you'll notice a string of characters separated by dots (periods). The last word in the domain name represents the 'Top Level Domain'. The second word in the domain name is known as a second level domain name (this is optional though and depends on the domain name)
      1. Example
         1. .com
         2. .edu
         3. .gov
         4. .co.uk
         5. .gov.uk
         6. .com.au
      2. First and second level domain name for .com.au
         1. .au 🡪 Top Level
         2. .com 🡪 Second Level
   2. These top level domains are controlled by the Internet Assigned Numbers Authority (IANA) in a root zone database which is essentially a database of all available top level domains. You can view this database by visiting:
      1. <https://www.iana.org/domains/root/db>
4. Domain Registrars
   1. Because all the names in a given domain name have to be unique there needs to be a way to organize this all so that domain name aren’t duplicated. This is where domain registrars come in.
   2. A registrar is an authority that can assign domain names directly under one or more top level domains. These domains are registered with InterNIC, a service of ICANN, which enforces uniqueness of domain names across the Internet. Each domain name becomes registered in a central database known as the WhoIS database.
   3. Popular domain registrars include:
      1. Amazon, GoDaddy.com,123-reg.co.uk etc
5. Common DNS Types
   1. SOA Records
   2. NS Records
   3. A Records
   4. CNAMES
   5. MX Records
      1. Used for mail
   6. PTR Records
      1. A reverse of A record 🡪 looking up a name against an IP address
6. Start of Authority Record (SOA)
   1. SOA Records store information about:
      1. The name of the server that supplied data for that zone.
      2. The administrator of that zone
      3. The current version of the datafile
      4. The default number of seconds for the time-to-live (TTL) file on resource records
7. NS stands for Name Server Records
   1. They are used by top level domain servers to direct traffic to the Content DNS server which contains the authoritative DNS records.
      1. Example:
      2. A user type Hellocloudgurus2019.com 🡪 the browser doesn’t know the IP address for this domain
      3. The browser goes to the top-level domain server (.com) and querying for the authoritative DNS records
      4. 🡪 query the NS records 🡪 the records give us the start of authority (SOA) 🡪 get the DNS records
   2. A Records
      1. An A Record is the fundamental type of DNS record and the 'A' in A record stands for 'Address'. The A Record is used by the computer to translate the name of the domain to an IP address. For example https://google.com -> <https://92.123.12.1>
8. TTL
   1. The length that a DNS record is cached on either the Resolving Server or the users own local PC is equal to the value of the 'Time To Live' (TTL) in seconds. The lower the time to live, the faster changes to DNS records take to propagate throughout the internet.
      1. Default (48H)
      2. DNS change take 48H to effect
9. CNames
   1. A Canonical Name (CName) can be used to resolve one domain name to another. For example, you may have a mobile website with a domain name http://m.acloud.guru that is used for when users browse to your domain name on their mobile devices. You may also want the name http://mobile.acloud.guru to resolve to this same address. (map one to another)
10. Alias Records
    1. Alias records are used to map resource record sets in your hosted zone to Elastic Load Balancers, CloudFront distributions, or S3 buckets that are configured as websites
    2. Alias records work like a CNAME record in that you can map one DNS name ([www.example.com](http://www.example.com)) to another ‘target’ DNS name (ellb1234.elb.amazonaws.com)
    3. Key difference with CNAME
       1. A CNAME can’t be used for naked domain names (zone apex record. 🡪 entire domain without a “www” in front) You can’t have a CNAME for <http://acloud.guru>, it must be either an A record or an Alias.
       2. Given the choice, always choose and Alias Record over a CNAME.

### Lab 51: Register A Domain Name

1. Category: Networking & Content Delivery 🡪 Route 53
2. You can buy domain names directly with AWS
3. It can take up to 3 days to register depending on the circumstances

## Routing Policies Available on AWS

1. Lab53: Simple Routing Policy
   1. If you choose the simple routing policy, you can only have one record with multiple IP addresses
   2. If you specify multiple values in a record, Route 53 returns all values to the user in a random order
   3. You can’t have a health check
2. Lab 54: Weighted Routing Policy
   1. Allows you to split your traffic based on different weights assigned.
   2. For example, you can set 10% of your traffic to go to US-EAST-1 and 90% to go to EU-WEST-1
   3. Health Checks
      1. You can set health checks on individual record sets
      2. If a record set fails a health check it will be removed from Route53 until it passes the health check
      3. You can set SNS notifications to alert you if a health check is failed
3. Lab 55: Latency-Based Routing
   1. Allows you to route your traffic based on the lowest network latency for your end user (ie which region will give them the fastest response time)
   2. To use latency-based routing you create a latency resource record set for the EC2 (or ELB) resource in each region that hosts your website. When Route 53 receives a query for your site, it selects the latency resource record set for the region that gives the user the lowest latency. Route 53 then responds with the value associated with that resource record set
4. Lab 56: Failover Routing Policy
   1. Failover routing policies are used when you want to create an active/passive set up. For example, you may want your primary site to be in EU-WEST-2 and your secondary DR site in AP-SOUTHEAST-2
   2. Route 53 will monitor the health of your primary site using a health check.
   3. A health check monitors the health of your endpoints.
5. Lab 57: Geolocation Routing Policy
   1. Geolocation routing lets you choose where your traffic will be sent based on the geographic location of your users (ie the location from which DNS queries originate).
   2. For example, you might want all queries from Europe to be routed to a fleet of EC2 instances that are specifically configured for your European customers. These servers may have the local language of your European customers and all prices are displayed in Euros.
6. Lab 58: Geoproximity Routing (Traffic Flow Only)
   1. Geoproximity routing lets Amazon Route 53 route traffic to your resources based on the geographic location of your users and your resources. You can also optionally choose to route more traffic or less to a given resource by specifying a value, known as a bias. A bias expands or shrinks the size of the geographic region from which traffic is routed to a resource
   2. To use geoproximity routing, you must use Route 53 traffic flow
7. Lab 59: Multivalue Answer Policy
   1. Multivalue answer routing lets you configure Amazon Route 53 to return multiple values, such as IP addresses for your web servers, in response to DNS queries. You can specify multiple values for almost any record, but multivalue answer routing also lets you check the health of each resource, so Route 53 returns only values for healthy resources.
   2. This is similar to simple routing however it allows you to put health checks on each record set.

# VPCs

## VPC - Virtual Private Cloud

1. What is a VPC?
   1. Virtual logical data center in the aws cloud
   2. Amazon Virtual Private Cloud (Amazon VPC) lets you provision a logically isolated section of the AWS Cloud where you can launch AWS resources in a virtual network that you define.
   3. You have complete control over your virtual network environment, including selection of your own IP address range, creation of subnets and config of route tables and network gateways.
      1. 1 Subnet = 1 Availability Zone
      2. We can have multiple subnets in the same availability zone, but one subnet can’t spread across multiple AZ
   4. You can easily customize the network config for your VPC. For example, you can create a public facing subnet of your webservers that has access to the internet and place your backend systems such as databases or application servers in a private-facing subnet with no internet access.
   5. You can leverage multiple layers of security, including security groups and network access control lists, to help control access to EC2 instances in each subnet.
      1. Security groups are Stateful; Network Access Control Lists are Stateless
      2. Security group only need to open up a port (80/22), it won’t worry about the outbound traffic (automatic)
      3. Network access control has to do both inbound and outbound
   6. Additionally, you can create a Hardware Virtual Private Network (VPN) connection between your corporate datacenter and your VPC and leverage the AWS cloud as an extension of your corporate datacenter.
   7. Use (cidr.xyz)[<https://cidr.xyz/>] to figure out subnet ranges within a VPC
   8. Consists of IGW(or Virtual Private Gateways), route tables, network access control lists (NACL), subnets, security groups
2. What can you do with a VPC?
   1. Launch instances into a subnet of your choosing
   2. Assign custom IP address ranges in each subnet
   3. Configure route tables between subnets
   4. Create single internet gateway and attach it to our VPC
   5. Much better security control over your AWS resources
   6. Instance security groups
   7. Subnet network access control lists (ACLS)
3. Default VPC vs Custom VPC
   1. Default VPC is user friendly, allowing you to immediately deploy instances
   2. All subnets in default VPC have a route to internet (all internet accessible)
   3. Each EC2 instance has both a public and private IP address.
4. VPC Peering
   1. Allows you to connect one VPC with another via a direct network route using private IP addresses
   2. Instances behave as if they are on the same private network.
   3. You can peer VPCs with other AWS accounts as well as with other VPCs in the same account
   4. Peering is in a star config: ie 1 central VPC peers with 4 others. NO TRANSITIVE PEERING!!
      1. A 🡪 B 🡨 C
      2. We can’t go through VPC from A to C though B, we have to do another peering connection directly between VPC A and C
         1. One to one basis
   5. You can peer between regions

### Lab 62: Build A Custom VPC

1. Category: Networking & Content Delivery
2. When you create a VPC a default Route Table, Network Access Control List (NACL) and a default Security Group
3. If won’t create any subnets, nor will it create a default internet gateway
4. US-East-1A in your AWS account can be a completely different availability zone to US-East-1A in another AWS account. The AZ’s are randomized
5. Amazon always reserve 5 IP addresses within your subnets
6. You can only have 1 internet gateway per VPC
7. Security Groups can’t span VPCs

### Lab 64: NAT Instances & NAT Gateways

1. NAT Instances - Network Address Translation
   1. When creating a NAT instance, Disable Source/Destination Check on the instance.
   2. NAT instances must be in a public subnet
   3. There must be a route out of the private subnet to the NAT, in order for this to work.
   4. The amount of traffic that NAT instances can support depends on the instance size. If you are bottlenecking, increase the instance size.
   5. You can create high availability using Autoscaling Groups, multiple subnets in different AZs, and a script to automate failover.
   6. Always behind a Security Group.
2. NAT Gateways
   1. Redundant inside the Availability Zone (one in one)
   2. Preferred by the enterprise
   3. Starts at 5Gbps and scales currently to 45Gbps
   4. No need to patch
   5. Not associated with security groups
   6. Automatically assigned public IP address
   7. No need to disable Source/Destination Checks
   8. If you have resources in multiple Availability Zones and they share one NAT gateway, in the event that the NAT gateway’s Availability Zone is down, resources in the other Availability Zones lose internet access. To create an Availability Zone-independent architecture, create a NAT gateway in each Availability Zone and configure your routing to ensure that resources use the NAT gateway in the same Availability Zone

### Lab 65: NACL - Network Access Control Lists

1. Your VPC automatically comes with a default NACL, and by default it allows all inbound and outbound traffic.
2. You can create custom NACLs. By default, each custom NACL denies all inbound and outbound traffic until you add rules
3. Each subnet in your VPC must be associated with a network ACL. If you don't explicitly associate a subnet with a network ACL, the subnet is automatically associated with the default NACL.
4. Block IP Addresses using network ACLs not Security Groups
5. You can associate a NACL with multiple subnets; however, a subnet can be associated with only one network ACL at a time. When you associate a NACL with a subnet, the previous association is removed.
6. NACLs contain a numbered list of rules that is evaluated in order, starting with the lowest numbered rule
   1. If you want to deny something you must put it in front of allow role
7. NACL have separate inbound and outbound rules, and each rule can either allow or deny traffic.
8. NACL are stateless; responses to allowed inbound traffic are subject to the rules for outbound traffic (and vice versa)

### Lab 66: Custom VPCs and ELBs

* 1. You need at least 2 public subnets in order to deeply and application load balancer

### Lab 67: VPC Flow Logs

1. What are VPC Flow Logs?
   1. VPC Flow Logs is a feature that enables you to capture info about the IP traffic going to and from network interfaces in your VPC. Flow log data is stored using Amazon CloudWatch Logs.
   2. After you've created a flow log, you can view and retrieve its data in Amazon CloudWatch Logs.
2. Flow logs can be created at 3 levels:
   1. VPC
   2. Subnet
   3. Network Interface Level
3. Exam Tips
   1. You cannot enable flow logs for VPCs that are peered with your VPC unless the peer VPC is in your account.
   2. You cannot tag a flow log
   3. After you've created a flow log, you cannot change its configuration; for example, you can’t associate a different IAM role with the flow log
   4. **Not all IP traffic is monitored**
      1. Traffic generated by instances when they contact the Amazon DNS server. If you use your own DNS server, then all traffic to the DNS server is logged.
      2. Traffic generated by a Windows instance for Amazon Windows license activation.
      3. Traffic to and from 169.254.169.254 for instance metadata
      4. DHCP traffic
      5. Traffic to the reserved IP address for the default VPC router

## Bastion Host

1. What is a bastion host?
   1. A **bastion host** is a special-purpose computer on a network specifically designed and configured to withstand attacks. The computer generally hosts a single application, for example a [proxy server](https://en.wikipedia.org/wiki/Proxy_server), and all other services are removed or limited to reduce the threat to the computer. It is hardened in this manner primarily due to its location and purpose, which is either on the outside of a [firewall](https://en.wikipedia.org/wiki/Firewall_(computing)) or in a demilitarized zone ([DMZ](https://en.wikipedia.org/wiki/Demilitarized_zone_(computing))) (public subnet) and usually involves access from untrusted networks or computers.
2. Exam Tips
   1. A NAT Gateway/Instance is used to provide internet traffic to EC2 instances in private subnets
      1. Not using SSH/RDP
   2. A Bastion is used to securely administer EC2 instances using SSH or RDP
      1. Bastions are called Jump Boxes in Australia
   3. You can’t use a NAT Gateway as a Bastion host

## Direct Connect

1. What is Direct Connect?
   1. AWS Direct Connect is a cloud service solution that makes it easy to establish a dedicated network connection from your premises to AWS. Using AWS Direct Connect, you can establish private connectivity between AWS and your datacenter, office, or colocation environment, which in many cases can reduce your network costs, increase bandwidth throughput, and provide a more consistent network experience than Internet-based connections.
2. Exam Tips:
   1. Direct Connect directly connects your data center to AWS
   2. Useful for high throughput workloads (ie lots of network traffic)
   3. Or if you need a stable and reliable secure connection

## VPC Endpoints

1. What is a VPC Endpoint?
   1. A VPC endpoint enables you to privately connect your VPC to supported AWS services and VPC endpoint services powered by PrivateLink without requiring an internet gateway, NAT device, VPN connection, or AWS Direct Connect connection. Instances in your VPC do not require public IP addresses to communicate with resources in the service. Traffic between your VPC and the other service does not leave the Amazon network.
      1. Don’t need to go over the internet
   2. Endpoints are virtual devices. They are horizontally scaled, redundant, and highly available VPC components that allow communication between instances in your VPC and services without imposing availability risks or bandwidth constraints on your network traffic.
2. There are two types of VPC endpoints
   1. Interface Endpoint
      1. An interface endpoint is an elastic network interface with a private IP address that serves as an entry point for traffic destined to a supported service.
   2. Gateway Endpoint
      1. Amazon S3
      2. DynamoDB

# HA Architecture

## Elastic Load Balancers

1. What is a load balancer?
   1. A physical or virtual device that balances the network load across multiple web servers
2. Types of Load Balancers
   1. Application Load Balancer (Intelligent)
      1. Best suited for load balancing of HTTP(S) traffic. They operate at Layer 7 (OSI) and are application aware. They are intelligent, and you can create advanced request routing, sending specified requests to specific web servers.
   2. Network Load Balancer (Performance)
      1. Best suited for load balancing of TCP traffic where extreme performance is required. Operating at the connection level (Layer 4), Network Load Balancers are capable of handling millions of requests per second, while maintaining ultra-low latencies.
      2. If you need a fixed IP address / you need ultra high performance
   3. Classic Load Balancer (OG, Legacy Load Balancer)
      1. Used to load balance HTTP(S) applications and use Layer 7-specific features, such as X-Forwarded and stick-sessions. You can use strict Layer 4 load balancing for applications that rely purely on the TCP protocol.
      2. Need really simple rooting 🡪 no any intelligent behind it 🡪 keep cost down
3. Load Balancer Error (504)
   1. Classic Load Balancers (ELB)
      1. If your application stops responding within the idle timeout period, the ELB responds with a 504 error (the gateway has timed out)
      2. This means that the application is having issues. This could be either at the Web Server layer or at the Database Layer (Internal Server Error type)
      3. Solution: Identify where the application is falling, and scale up or out where possible
         1. Web Server or Database Server?
4. X-Forwarded-For Header
   1. If you need the IPv4 address of your end user (public IP address)

### Lab 36: Load Balancers

1. Instances are monitored but ELB are reported as InService or OutofService
2. Health Checks check the instance health by talking to it.
3. ELB's have their own DNS name. You are **never** given an IP address
4. Read the ELB FAQ for Classic Load Balancers

## Advanced Load Balancer Theory

1. What are sticky sessions?
   1. Classic Load Balancer routes each request independently to the registered EC2 instance with the smallest load
   2. Sticky sessions allow you to bind a user’s session to the same EC2 instance. This ensures that all requests from the user during the session are sent to the same instance
      1. Can be useful if you are storing information locally to that instance
   3. You can enable Sticky Sessions for Application Load Balancers as well, but the traffic will be sent to at the Target Group Level
2. What is Cross Zone Load Balancing?
   1. Cross Zone Load Balancing enables you to load balance across multiple availability zone
3. What are path patterns?
   1. Path patters allow you to direct traffic to different EC2 instances based on the URL contained in the request
   2. You can create a listener with rules to forward requests based on the URL path. This is known as path-based routing. If you are running microservices, you can route traffic to multiple back-end services using path-based routing. For example, you can route general requests to one target group and requests to render images to another target group

### Lab 75: Launch Configurations & Auto Scaling Groups

## HA Architecture

1. Exam Tips:
   1. You should always plan for failure
   2. Use multiple AZ’s and multiple regions where ever you can
   3. Know the difference between Multi-AZ and Read Replicas for RDS
      1. Multi-az 🡪 recovery
      2. Read replicas 🡪 performance
   4. Know the difference between scaling out and scaling up
      1. Scaling out 🡪 auto scaling groups 🡪 add additional EC2 instances
      2. Scaling up 🡪 increase the resources inside EC2 instances (amount of RAM/CPE)
   5. Read the question carefully and always consider the cost element
   6. Know the difference S3 storage classes

### Lab 77-80: Building a fault tolerant WordPress site

1. Getting Setup
2. Setting Up EC2
3. Adding Resilience & Autoscaling
4. Cleaning Up
5. CloudFormation
   1. Is a way of completely scripting your cloud environment
   2. Quick Start is a bunch of CloudFormation templates already built by AWS Solutions Architects allowing you to create complex environments very quickly

### Lab 82: Elastic Beanstalk

1. With Elastic Beanstalk, you can quickly deploy and manage applications in the AWS Cloud without worrying about the infrastructure that runs those applications. You simply upload your application, and Elastic Beanstalk automatically handles the details of capacity provisioning, load balancing, scaling, and application health monitoring

# Application

## SQS - Simple Queue Service

1. What is SQS?
   1. Amazon SQS is a web service that gives you access to a message queue that can be used to store messages while waiting for a computer to process them.
   2. Amazon SQS is a distributed queue system that enables web service applications to quickly and reliably queue messages that one component in the application generates to be consumed by another component. A queue is a temporary repository for messages that are awaiting processing.
   3. SQS is a way to de-couple your infrastructure
   4. SQS is a pull based, not pushed based
   5. SQS quarantees that your messages will be processed at least once
2. SQS Breakdown
   1. Using Amazon SQS, you can decouple the components of an application, so they run independently easing message management between components
   2. Any component of a distributed application can store messages in the queue. Messages can contain up to 256Kb of text in any format. Any component can later retrieve the messages programmatically using the SQS API
   3. Manages can be kept in the queue from 1 minute to 14 days; the default retention period is 4 days
3. What do you mean by "Queue"?
   1. The queue acts as a buffer between the component producing and saving data, and the component receiving the data for processing. This means the queue resolves issues that arise if the producer is producing faster than the consumer can process it, of if the producer or consumer are only intermittently connected to the network.
4. Queue Types
   1. Standard Queue (default)
      1. Amazon SQS offers standard as the default queue type. A standard queue lets you have a nearly-unlimited number of transactions per second. Standard queues guarantee that a message is delivered at least once. However, because of the highly distributed architecture that allows high throughput, more than one copy of a message might be delivered out of order. Standard queues provide best effort ordering which ensures that messages are generally delivered in the same order as they are sent.
      2. Standard order is not guaranteed, and messages can be delivered more than once
   2. FIFO Queues (First In, First Out)
      1. The FIFO queue complements the standard queue. The most important features of this queue type are FIFO delivery and exactly one processing: The order in which messages are sent and received is strictly preserved and a message is delivered once and remains available until a consumer processes and deletes it; duplicates are not introduced into the queue.
      2. FIFO queues also support message groups that allow multiple ordered message groups within a single queue. FIFO queues are limited to 300 transactions per second, but have all the capabilities of standard queues
      3. |\_5\_| ---> |\_4\_| ---> |\_3\_| ---> |\_2\_| ---> |\_1\_|
      4. FIFO order is strictly maintained, and messages are delivered only once
5. Visibility Time Out
   1. Visibility Time Out is the amount of time that the message is invisible in the SQS queue after a reader picks up that message. Provided the job is processed before the visibility time out expires, the message will then be deleted from the queue. If the job is not processed within that time, the message will become visible again and another reader will process it. This could result in the same message being delivered twice
      1. Visibility timeout maximum is 12 hours
6. Long polling
   1. Amazon SQS long polling is a way to retrieve messages from your Amazon SQS queues. While the regular short polling returns immediately (even if the message queue being polled is empty), long polling doesn’t return a response until a message arrives in the message queue, or the long poll times out

## SWF - Simple Work Flow Service

1. What is SWF?
   1. Amazon Simple Workflow Service is a web service that makes it easy to coordinate work across distributed application components. Amazon SWF enables applications for a range of use cases, including media processing, web application back-ends, business process workflows, and analytics pipelines, to be designed as a coordination of tasks.
   2. Tasks represent invocations of various processing steps in an application which can be performed by executable code, web service calls, human actions, scripts.
2. SWF vs SQS
   1. SQS has a retention period of up to 14 days; with SWF, workflow executions can last up to 1 year
   2. Amazon SWF presents a task-oriented API, whereas Amazon SQS offers a message-oriented API
   3. Amazon SWF ensures that a task is assigned only once and is never duplicated. With Amazon SQS, you need to handle duplicated messages may also need to ensure that a message is processed only once.
   4. Amazon SWF keeps track of all the tasks and events in an application. With Amazon SQS, you need to implement your own application-level tracking, especially if your application uses multiple queues.
3. SWF Actors
   1. Workflow Starters
      1. An application that can initiate (start) a workflow. Could be your e-commerce website when placing an order or a mobile app searching for bus times
   2. Deciders
      1. Control the flow of activity tasks in a workflow execution. If something has finished (or failed) in a workflow, a Decider decides what to do next
   3. Activity Workers
      1. Carry out the activity tasks

## SNS - Simple Notification Service

1. What is SNS?
   1. SNS is a web service that makes it easy to set up, operate and send notifications from the cloud.
   2. It provides developers with a highly scalable, flexible and cost-effective capability to publish messages from an application and immediately deliver them to subscribers or their applications
   3. Push notifications to Apple, Google, Fire OS and Windows devices as well as Android devices in China with Baidu Cloud Push.
   4. Besides pushing cloud notifications directly to mobile devices, SNS can also deliver notifications by SMS text message or email, to SQS queues, or to any HTTP endpoint.
2. What is a topic?
   1. SNS allows you to group multiple recipients using topics. A topic is an "access point" for allowing recipients to dynamically subscribe for identical copies of the same notification.
   2. One topic can support deliveries to multiple endpoint types - for example, you can group together iOS, Android and SMS recipients. When you publish once to a topic, SNS delivers appropriately formatted copies of your message to each subscriber.
3. SNS Availability
   1. To prevent messages from being lost, all messages published to SNS are stored redundantly across multiple availability zones.
4. SNS Benefits
   1. Instantaneous, push-based delivery (no polling)
   2. Simple APIs and easy integration with applications
   3. Flexible message delivery over multiple transport protocols
   4. Inexpensive, pay-as-you-go model with no up-front costs
   5. Web-based AWS Management Console offers the simplicity of a point-and-click interface
5. SNS vs SQS
   1. Both messaging services in AWS
   2. SNS = push; SQS = polls (pulls)

## Elastic Transcoder

1. What is Elastic Transcoder?
   1. Media Transcoder in the cloud.
   2. Convert media files from their original source format in to different formats that will play on smartphones, tablets, PCs etc.
   3. Provides transcoding presets for popular output formats, which means that you don't need to guess about which settings work bets on particular devices.
   4. Pay based on the minutes that you transcode and the resolution at which you transcode.

## API Gateway

1. What is API Gateway?
   1. API Gateway is a fully managed service that makes it easy for developers to publish, maintain, monitor and secure APIs at any scale.
   2. With a few clicks in the AWS Management Console, you can create and API that acts as a "front door" for applications to access data, business logic, or functionality from you back-end services, such as applications running on EC2, code running on Lambda or any web application.
2. What can API Gateway do?
   1. Expose HTTPS endpoints to define a RESTful API
   2. Serverless-ly connect to services like Lambda & DynamoDB
   3. Send each API endpoint to a different target
   4. Run efficiency with low cost
   5. Scale effortlessly
   6. Track and control usage by API key
   7. Throttle requests to prevent attacks
   8. Connect to CloudWatch to log all requests for monitoring
   9. Maintain multiple versions of your API
3. How do I configure API Gateway?
   1. Define an API (container)
   2. Define Resources and nested Resources (URL paths)
   3. For each Resource:
      1. Select supported HTTP methods (verbs)
      2. Set security
      3. Choose target (such as EC2, Lambda, DynamoDB, etc)
      4. Set request and response transformations
4. How do I deploy API gateway?
   1. Deploy API to a stage:
      1. Uses API Gateway domain, by default
      2. Can use custom domain
      3. Now supports AWS Certificate Manager: free SSL/TLS certs
5. API Gateway Caching
   1. You can enable API caching in API Gateway to cache your endpoints response. With caching, you can reduce the number of calls made to your endpoint and also improve the latency of the requests to your API.
   2. When you enable caching for a stage, API Gateway caches responses from your endpoint for a specified TTL period, in seconds. API Gateway then responds to the request by looking up the endpoint response from the cache instead of making a request to your endpoint.
6. Same Origin Policy
   1. In computing, the same-origin policy is an important concept in the web application security model. Under the policy, a web browser permits scripts contained in a first web page to access data in a second web page, but only if both web pages have the same origin.
   2. This is done to prevent Cross-Site Scripting (XSS) attacks
      1. Enforced by web browsers
      2. Ignored by tools like PostMan and curl
7. CORS (Cross-origin resource sharing)
   1. CORS is one way the server at the other end (not the client code in the browser) can relax the same-origin policy.
   2. CORS is a mechanism that allows restricted resources (fonts) on a web page to be requested from another domain outside the domain from which the first resource was served
   3. CORS in Action
      1. Brower makes an HTTP OPTIONs call for a URL (OPTIONS is an HTTP method like GET, PUT, and POST)
      2. Server returns a response that says:
         1. “These other domains are approved to GET this URL”
      3. Error – “Origin policy can’t be read at the remote resource?” You need to enable CORS on API Gateway
8. Exam Tips:
   1. API Gateway has caching capabilities to increase performance
   2. API Gateway is low cost and scales automatically
   3. You can throttle API Gateway to prevent attacks
   4. You can log results to CloudWatch
   5. If you are using JavaScript/AJAX that uses multiple domains with API Gateway, ensure that you have enabled CORS on API Gateway
   6. CORS is enforced by the client

## Kinesis

1. What is streaming data?
   1. Streaming data is data that is generated continuously by thousands of data sources, which typically send in the data records simultaneously, and in small sizes (order of KB)
      1. Purchases from online stores (amazon.com)
      2. Stock prices
      3. Game data
      4. Social network data
      5. Geospatial data - uber, google maps
      6. iOT data
2. What is Kinesis?
   1. AWS Kinesis is a platform on AWS to send your streaming data to. Kinesis makes it easy to load and analyze streaming data, and also providing the ability for you to build your own custom applications for your business needs.
3. Types of Kinesis
   1. Kinesis Streams
      1. Streams consist of shards; data is contained in shards.
      2. 5 transactions per second for reads, up to a maximum total data read rate of 2Mb per second and up to 1,000 records per second for writes, up to a maximum total data write rate of 1 Mb per second (including partition keys).
      3. The data capacity of your stream is a function of the number of shards that you specify for the stream. The total capacity of the stream is the sum of the capacities of its shards.
   2. Kinesis Firehose
      1. No persistent storage (no shards)
         1. You need to do something with the input data
      2. Lambda functions inside
   3. Kinesis Analytics
      1. Works with both kinesis above

## Web Identity Federation & Cognito

1. What is web identity federation?
   1. Web Identity Federation lets you give your users access to AWS resources after they have successfully authenticated with a web-based identity provider like Amazon, Facebook, or Google. Following successful authentication, the user receives an authentication code from the Web ID provider, which they can trade for temporary AWS security credentials allowing them to assume an IAM role
2. Amazon Cognito
   1. Amazon Cognito provides web identity federation with the following features:
      1. Sign-up and sign-in to your apps
      2. Access for guest users
      3. Acts as an identity broker between your application and web id providers, so you don’t need to write any additional code
      4. Synchronizes user data for multiple devices
      5. Recommended for all mobile applications AWS services
   2. Use Cases
      1. Cognito brokers between the app and Facebook or Google to provide temporary credentials which map to an IAM role allowing access to the required resources
      2. No need for the application to embed or store AWS credentials locally on the device and it gives users a seamless experience across all mobile devices
3. Cognito Pools
   1. User Pools are user directories used to manage sign-up and sign-in functionality for mobile and web applications. Users can sign-in directly to the User Pool, or using Facebook, Amazon, or Google. Cognito acts as an Identity Broker between the identity provider and AWS. Successful authentication generates a JSON Web token (JWTs)
      1. User pool is user based. It handles things like user registration, authentication, and account recovery.
   2. Identity Pools enable provide temporary AWS credentials to authorize access AWS services like S3 or DynamoDB
4. Cognito Synchronization
   1. Cognito tracks the association between user identity and the various different devices they sign-in from. In order to provide a seamless user experience for your application, Cognito uses Push Synchronization to push updates and synchronize user data across multiple devices. Cognito uses SNS is to send a notification to all the devices associated with a given user identity whenever data stored in the cloud changes

# Serverless

## Lambda

1. What is Lambda?
   1. AWS Lambda is a compute service where you can upload your code and create Lambda function. AWS Lambda takes care of provisioning and managing the servers that you use to run the code. You don’t have to worry about operating systems, patching, scaling, etc.
2. You can use Lambda in the following ways:
   1. As an event-driven compute service where AWS Lambda runs your code in response to events. These events could be changes to data in an Amazon S3 bucket or an Amazon DynamoDB table.
   2. As a compute service to run your code in response to HTTP requests using Amazon API Gateway or API calls made using AWS SDKs.
3. **Encapsulation of the following (Lambda is the ultimate extraction layer):**
   1. Data Centers
   2. Hardware
   3. Assembly Code/Protocols
   4. High Level languages
   5. Operation Systems
   6. Application Layer/AWS API's
   7. AWS Lambda
4. Compatible Languages:
   1. C#
   2. Java
   3. Node.js
   4. Python
   5. Go
   6. PowerShell
5. How is Lambda priced?
   1. Number of requests
      1. First 1m request are free. $0.20 per 1m requests thereafter.
   2. Duration
      1. Duration is calculated from the time your code begins execution until it returns or otherwise terminates, rounded up to the nearest 100ms. The price depends on the amount of memory you allocate to your function. You are charged $0.00001667 for every GB-second used.
6. Why is Lambda cool?
   1. No SERVERS!!
   2. Continuous Scaling
   3. Super cheap
7. Exam Tips
   1. Lambda scales out (not up) automatically
   2. Lambda functions are independent, 1 event = 1 function
   3. Lambda is serverless
      1. Know what services are serverless
         1. S3
         2. API Gateway
         3. DynamoDB
      2. Not serverless
         1. RDS
            1. Only Raw RDS service is surplus
         2. EC2
   4. Lambda functions can trigger other lambda functions, 1 event can = x functions if functions trigger other functions
   5. Architectures can get extremely complicated; AWS X-ray allows you to debug what is happening
   6. Lambda can do things globally; you can use it to back up S3 buckets to other S3 buckets etc
   7. Know your triggers – connecting AWS services

### Lab 93: Build a serverless webpage with API Gateway & Lambda

### Lab 94: Build an Alexa Skill