Session 1: AWS Storage-Types of Storage

Types of Storage :

Simple Storage Service(S3) : Object Level Storage and access it from anywhere
Elastic File System(EFS) : Is only for Linux
Elastic Block Storage(EBS) : Block level storage which can only be accessed through EC2 to it is attached
Glacier : Is recently called as S3 Glacier, which mostly used to store data which is not most imp but we want to store for future use
Snowball : Uses for data migration consists of huge data, which is a portable device sent usually in trucks to data centers of company which wants to migrate to cloud

- AWS offers a complete range of storage services to support both application and archival compliance requirements.Select from objects,file and block storage services as well as cloud data migration options to start designing the foundation of our cloud IT Environment

Session 2 : Block Storage vs Object Storage

Block Storage :

- It is suitable for transactional databases,random read/write loads and structured database storage
- Block storage divide the data to be stored in evenly sized blocks(data chunks) for instance ,it can split into evenly sized blocks before it is stored
- Data blocks stored in block storage would not contain metadata(data created,data size,data modified,content type etc)
- Block storage only keeps the address(index) where the data blocks are stored,it does not care what is in that block.Just now how to retrive it when required
- EBS Volume : Can only be accessed through EC2

Object Storage :

- It stores the files as a whole and does not divide them
- In this,an object is the file/data itself,its metadata,object global unique id
- The Object global unique id is a unique identifier for the object(can be the object name itself) and it must be unique such that it can be retrived disregarding where its physical storage location is
- Object storage cannot be mounted as a drive
- It is accessed from the Internet
- Ex : AWS S3,Dropbox etc

Session 3 : Simple Storage Service(S3),Naming Rules,Sub Resources

Simple Storage Service (S3) :

- S3 is a storage for the internet.it has a simple webservices interface for simple storing & retrieving of any amount of data,anytime from anywhere on the internet
- S3 is object based storage
- We cannot install OS on S3
- S3 has a distributed data-store architecture where objects are redundantly stored in multiple locations(Min 3 Location in same region)
- Data is stored in Buckets
- A bucket is a flat container of objects
- Max capacity of a bucket is 5TB

•We can create folder in our bucket(Available through Console)
- We cannot create nested buckets
- Bucket ownership is non-transferrable
- S3 bucket is region specific
- We can have upto 100 Buckets per account (may expand on request)

S3 Bucket Naming Rules :

- S3 bucket names(keys) are globally unique across all AWS regions
- Bucket names cannot be changed after they are created
- If a bucket is deleted,its name becomes available again to us or other accounts to use
- Bucket names must be atleast 3 and no more than 63 characters
- Bucket names are part of the URL used to access a bucket
- Bucket name must be a series of one or more lables(xyz.Bucket)
- Bucket names can contain lowercase,numbers and hypen(-).We cannot use uppercase letters
- Bucket name should not be an IP Address
- Each label must start and end with a lowercase letter or a number
- By default buckets and its objects are private
- By default only owner can access the bucket

S3-Bucket Subresources :

- This Includes :

1.Lifecycle -To decide an objects lifecycle management
2.Website -To hold configurations related to static website hosted S3 Bucket
3.Versioning -Keep object versions as it changes (Gets Updated) ....Can be enabled or suspended but can not be disabled after enabling
4.Access Control List : Bucket Policies

- The name is simply two parts-Bucket Region's endpoint and bucket name

Eg : S3 bucket named mybucket in europe west region is https://s3-eu-west1.amazonaws.com/-mybucket

S3 Objects :

- An object size stored in an S3 bucket can be 0Byte to 5TB
- Each object is stored and retrived by unique key(ID or Name)
- An object in AWS S3 is uniquely identified and addressed through -service endpoint,bucket name,object key(name),optionally object versions
- objects stored in a S3 bucket in a region will never leave that region unless we specifically move them to another region or CRR
- A bucket owner can grant cross account permissions to another aws account(or users in another account) to upload objects
- We can grant S3 Bucket/object permissions to Individual users,AWS Account,Make the resource public,to all authentic users

Session 4 : S3Bucket Versionning,MFA Delete,S3 Multipart Upload,Copying S3 Objects

S3 Bucket Versioning :

- Bucket versioning is a S3 bucket sub-resource used to protect against accidental object/data deletion or overwriten
- Versioning can also be used for data retention or archive
- Once we enable versioning on a bucket,it cannot be disabled however it can be suspended
- When enabled,bucket versioning will protect existing object and new object and maintains their versions as they are updated
- Updating objects refers to PUT,POST,COPy,Delete actions on objects
- When versioning is enabled and we try to delete an object a delete marker is placed on the

object
- We can still view the object and the delete marker
- If we reconsider deleting the objects,we can delete the “Delete marker” and the object will be available again
- We will be charged for all S3 storage cost for all object versions stored
- We can use versioning with S3 lifecycle policies to delete older versions or we can move them to a cheaper S3 storage(Or Glacier)
- Bucket Version States : Enabled,Suspended,Un-versioned
- Versioning applies to all objects in a bucket and not partially applied
- Object existing before enabling versioning will have a version ID as ‘Null’
- If we have a bucket that is already versioned,then when we suspend versioning existing objects and their versions remain as it is
- However they will not be updated/versioned further with future updates while the bucket versioning is suspended
- New objects(uploaded after suspension), they will have a version ID as ‘Null’
- If the same key(name) is used to store another objects,it will override the existing one
- An object deletion in a suspended versioning buckets will only delete the objects with ID ‘Null’

S3 Bucket Versioning - MFA Delete :

- MFA delete is a versioning capacity that adds another layer of security in case our account is compromised
- This adds another layer of security for changing our buckets versioing state and permanently deleting an object version
- MFA delete requires our security credentials and the code displayed on an approved physical or software based authentication device

S3 Multipart Upload :

- Is used to upload an object in parts
- parts are uploaded independently and in parallel in any order
- It is recommended for object sizes 100Mb or more as min size is 5MB to multiport upload
- We must use it for objects larger than 5GB
- This is done through S3 multipart upload API

Copying S3 Objects :

- The copy operation creates a copy of an object that is already stored in Amazon S3
- We can create a copy of our object upto 5Gb in size in a single atomic operations
- However to copy an object greater than 5GB,we must use the multipart upload API
- Incur charges if copy to another region
- Use the copy operation to generate additional copies of the subject,renaming object(copy to a new name),Changing the copy's storage class or encrypt it at rest
- Move object across AWS location/region
- Change object metadata

Session 5 : Storage Classes of Amazon S3

Storage Classes of Amazon S3 :

1.S3-Standard(for normal and very frequent access),
2.S3 Glacier Deep Archive(Cheapest),
3.Amazon Glacier(Long Term Storage),
4.S3 Standard Infrequent Access(Less cost but we pay to access it more frequently),-Standard IA
5.S3 One-Zone-IA(Only stores1 copy with less cost),
6.S3 Intelligent Tiering(Automatically shifts data between standard, standard IA,glacier etc based on usage)
7.S3 Reduced Redundancy Storage (was removed)

Amazon S3 Standard :

- It offers high durability availability and performance object storage for frequently accessed data
- Durability is 99.999999999% (11 time 9)
- Resilient against events that impact entire AZ
- Designed for 99.99% availability over a given year
- Support SSL for data in-transit and encryption of data at rest
- Storage costs for the object is fairly high but there is very less charge for accessing the objects
- Largest object that can be uploaded in a single put is 5GB
- Backed wih the Amazon S3 service level Agreement for availability

Amazon S3-IA :

- S3-IA is for data that is accessed less frequently but required rapid access when needed
- The storage cost is much cheaper than S3-Standard,almost half the price.But we are charged more heavily for accessing our objects
- Durability is 99.999999999%
- Resilient against events that impact entire AZ
- Availability is 99.9% in year
- Support SSL for data in transit and encryption of data at rest
- Data that is deleted from S3-IA within 3 days will be charged for a full 30 days
- Backed wih the Amazon S3 service level Agreement for availability

Amazon S3 Intelligent Tiering :

- This storage class is designed to optimize cost by automatically moving data to the most cost effective access-tier
- It works by storing objects in two access tiers
- If an object in the infrequent access tier is accessed,it is automaticaly moved back to the frequent access tier
- There are no retrieval fess when using the S3-Intelligent tiering storage class and no additional tiering fee when objects are moved between access tiers
- Resilient against events that impact entire AZ
- Same low latency and high performance of S3-Standard
- Object less than 128Kb cannot move to IA
- Durability is 99.999999999%
- Availability is 99.9%
- Backed wih the Amazon S3 service level Agreement for availability

Amazon S3 One-Zone IA :

- It is for data that is accessed less frequently but require rapid access when needed
- Data stored in single AZ
- Ideal for those who want lower cost option of IA-data
- It is good choice for storing secondary backup copies of on-premise data or easily recreatable data
- We can use S3 lifecycle policies
- Durability is 99.999999999%
- Availability is 99.5%
- Because S3 one zone-IA stores data in single Az,data stored in this storage class will be lost in the event of AZ destruction
- Backed wih the Amazon S3 service level Agreement for availability

Amazon S3 Glacier :

- S3 glacier is a secure,durable,low cost storage class for data archiving
- To keep cost low yet suitable for varyig needs,S3 glacier provides three retrieval options that range from a few minutes to hours
- We can upload object directly to glacier or use lifecycle policies
- Durability is 99.999999999%
- data is resilient in the event of one entire AZ destruction

- Support SSL for data in transit and encryption of data at rest
- We can retrieve 10GB of our amazon S3 glacier data per month for free with free tier account
- Backed wih the Amazon S3 service level Agreement for availability

Amazon S3 Glacier Deep Archieve :

- It is amazon S3's cheapest storage
- Design to retain data for long period Eg : 10Years
- All objects stored in S3-Glacier deep archieve are replicated and stored across atleast at three geographically dispersed AZ
- Durability is 99.999999999%
- Ideal alternative to magnetic tape libraries
- Retrieval time within 12hours
- Storage cost is upto 75% less than for the existing S3 glacier storage class
- Availability is 99.9%
- Backed wih the Amazon S3 service level Agreement for availability

LABS :

Session 6 : S3 Bucket Creation

Session 7 : Bucket Creation using CLI

Session 8 : Versioning

Session 9 : Enable Cross Region Replication/Data Replication

Cross Region Replication :

- CRR enables automatic,asynchronous copying of objects across buckets in diff aws regions.Buckets configured for cross regions replication can be owned by the same aws account or by diff accounts
- CRR is enabled with a bucket level configuration.We add the replication configuration to our source bucket.
- In the min configuration,we provide the destination bucket,where we want amazon S3 to replicate objects and an aws IAM role that amazon s3 can assume to replicate objects on our behalf

When to use CRR :

- Comply with compliance requirements
- Minimize latency
- Increase operational efficiency
- Maintain object copies under diff ownership

Session 10 : AWS S3 Object Lifecycle Management

Session 11 : Elastic File System

Elastic File System :

- It is fully managedd service that makes it easy to set up,scale and cost-optimize file storage in the Amazon Cloud.
- With few clicks in the aws mgt. console, we can create file systems that are accessible to amazon ec2 instances via a file system interface(using standard OS file I/O API's) and support full file system access semantics (such as strong consistency and file locking)
- These file systems can automatically scale from gigabytes to petabytes of data without needing

to provision storage.
- Tens,hundereds or even thousands of amzon ec2 instances can access an amazon efs file system at the same time and amazon efs provides consistent performance to each amazon ec2 instance
- Amazon efs is designed to be highly durable and highly available
- There are no min fee or setup costs,we pay only for what we use
- This is only for Linux machines
- Designed to provide performance for a broad spectrum of workloads and applications,including big data and analytics,media processing workflows,content mgt.,web serving and home directories
- This is not available in all regions

Session 12 : Hosting Static Website on S3

Session 13 : Elastic Block Storage(Diff. between EBS & Instance Store)

- Two types of block store devices are avaiabe for ec2 :

1.Elastic Block Storage : Persistent and Network attached drive

- Ebs voume behave like raw,unformatted external block storage devices that we can attach to our ec2 instance
- Ebs volumes are block storage devices suitable for database style data that require frequent read & write
- Ebs volumes are attached to our ec2 instances through the aws network,like Virtual hard drives
- An EBS volume can attach to a single ec2 instance only at a time
- Both ebs volumes and ec2 instance must be in the same AZ
- An ebs volume data is replicated by aws accross multiple servers in the same az to prevent data loss resulting from any single aws component failure

2.Instance Store Backed EC2 : Basically the virtual hard drive on the host allocated to this EC2 instance is limited to 10GB per device,ephemeral storage(non-persistent storage) and the ec2 instance can't be stopped(can only be rebooted) or terminated and terminate and stop will delete data

Session 14 : EBS Types-GP2,PIOPS,SSD,ST1,SC1 and Magnetic

EBS Volume Types :

1.SSD Backed Volume : General Purpose SSD(GP2) which is default volume and Provisioned IOPS SSD(io1)...These are bootable
2.HDD Backed Volume : Throughput Optimized HHD(st1) and cold HDD(sc1) ......These are non-bootable
3.Magnetic Standard : This is Bootable

General Purpose SSD(gp2) :

- GP2 is the default EBS volume type for the amazon ec2 instances and are backed by ssd
- GP,balances both price and performance
- Ratio of 3IOPS/GB with upto 10,000 IOPS(Input/Output per second)
- Boot volume having low latency
- Volume size 1Gb to 16TB
- Price : $0.10 per GB/Month

Provisioned IOPS SSD (io1) :

- These volumes are ideal for both IOPS intensive and throughput intensive workloads that

require extremely low latency or for mission critical application
- Designed for I/O intensive applications such as large relational or Nosql Databases
- Use if we need more than 10,000 IOPS
- Can provision upto 32000 IOPS per volume and 64,000 IOPS for nitro based instances
- Volume Size : 4Gb to 16Tb
- Price : $0.125 per GB/month

Throughput Optimized HDD(st1):

- ST1 is backed by hard disk drives and is ideal for frequently accessed,throughput intensive workloads with large datasets
- ST1 volumes deliver performance in term of throughput,measured in MB/s
- Big data,data warehouse,log processing
- It cannot be a boot volume
- Can provisioned upto 500 IOPS per volume
- Volume Size : 500GB to 16TB
- Price : $0.045 per GB/month

Cold HDD(SC1) :

- SC1 is also backed by HDD and provides the lowest cost per GB of all EBS volume Types
- Lowest cost storage for infrequent access workloads
- Used in file servers
- Cannot be a boot volume
- Can provisioned upto 250 IOPS per volume
- Voluem Size : 500GB to 16TB
- Price : $0.025 per GB/month

Magnetic Standard :

- Lowest cost per GB of all EBS volume type that is bootable
- Magnetic volumes are ideal for workloads where data is accessed infrequently and application where the lowest storage cost is important
- Price : $0.05 per GB/month
- Volume Size : 1GB to 1TB
- Max IOPS/Volume : 40 to 200

Session 15 : EBS Snapshots of root volume and non-root volume

EBS Snapshots :

- Ebs Snapshots are point-in-time images/copies of our ebs volume
- Any data written to the volume after the snapshot process is initiated,will be included in the resulting snapshot(but wiill be included in future,incremental update)
- Per aws account,upto 5000 ebs volumes can be created
- Per account,upto 10,000 EBS snapshots can be created
- EBS snapshots are stored in S3,however we cannot access them directly,we can only access them through ec2 API's
- While EBS volumes are AZ specific,snapshots are region specific
- Any AZ in region can use snapshot to create EBS volume
- To migrate on EBS from one AZ to another,create a snapshot(region specific) and create an EBS volume from the snapshot in the intended AZ
- We can create a snapshot to an EBS volume of the same or larger size than the original volume size from which the snapshot was initially created
- We can take a snapshot of a non-root ebs volume while the volume is in use of a running ec2 instance
- This means we can still access it while the snapshot is being processed
- However the snapshot will only include data that is already written to our volume
- The snapshot is created immediately but it may stay in pending status until the full snapshot is

completed.This may take few hours to complete specifically for the first time snapshot of a volume
- During the period,when the snapshot status is pending,we can still access the volume(non-root),but I/O might be slower because of the snapshot activity
- While in pending state,an in-progress snapshot will not include data from ongoing reads and writes to the volume
- To take complete snapshot of our non-root EBS volume : stop or unmount the volume
- To create a snapshot for a root ebs volume, we must stop the instance first then take the snapshot

Session 16 : Incremental Snapshots

Incrementa Snapshot:

- EBS snapshots are stored incrementally
- For low cost storage on s3,and a guarantee to be able to fully restore data from the snapshots
- What we need is a single snapshot that further snapshot will only carry the changed blocks(incremental updates)
- Therefore we do not need to have multiple full/complete copies of the snapshot
- We are charged for : data transferred to S3 from our ebs volume we are taking snapshots
- Snapshots stored in s3
- First snapshot is a clone,subsequent snapshotsa are incremental
- Deleting snapshot will only remove data exclusive to that snapshot

Session 17 : Encryption of EBS Volume

EBS Encyption :

- EBS encryption is supported on all ebs volume types and all ec2 instance familes
- Snapshots of encrypted volumes are also encrypted
- Creating an ebs volumes from an encrypted snapshot will result in an encrypted volume
- Data encryption at rest means encrypting data while it is stored on the data storage device

- There are many ways we can encrypt data on an ebs volume at rest,while the volume is attached to an ec2 instance :

1. using 3rd party encryption technique on ebs volume,
2. encryption tools,
3.using encrypted ebs volumes
4.using encryption at the OS Level
5.Encrypt data at the application level before storing it to the volume
6.Using encrypted file system on the top of the ebs volume
- Encrypted volumes are accessed exactly like unencrypted ones,basically encryption is handled transparently
- We can attach an encrypted and unencrypted volumes to the same ec2 instance
- Remember that the ebs volumes are not physically attached to the ec2 instance,rather they are virtually attached through the ebs infrastructure
- This means when we encrypt data ona an ebs volume,data is actually encrypted on the ec2 instance then transferred,encrypted to be stored on the ebs volume
- This means data in transit between ec2 and encrypted ebs volume in also encrypted
- There is no direct way to change the encryption state of the volume

- To change the state(indirectly) we need to follw either of the followig two ways :

1.Attach a new,encrypted ebs volume to the ec2 instance that has the data to be encrypted then:

□ Mount the new volume to the ec2 instance

□ Copy the data from the unencrypted volume to the new volume

□ Both volumes must be on the same ec2 instance

2.Create a snapshot of the encrypted volume

□ Copy the snapshot and choose encryption for the new copy,this will create an encrypted copy of the snapshot

□ Use this new copy to create on ebs volume which will be encrypted too

□ Attach the new encrypted ebs volume to the ec2 instance

Root EBS volume Encryption :

- There is no direct way to change the encryption state of a volume
- There is an indirect workaround to this

□ Launch the instance with ebs volume required

□ Do whatever patching or install applications

□ Create an AMI from the ec2 instance

□ Copy the AMI from the EC2 instance

□ Copy the AMI and choose encryption while copying

□ This results it an encrypted AMI that is private

□ Use the encrypted AMI to launch new ec2 instances which will have these ebs root volume encrypted

Session 18 : Sharing EBS Snapshots

EBS Encryption Key :

- To encrypt a volume or a snapshot we need an encryption key,these keys are called customer masters key(CMK) and are managed by aws key mgt. service(KMS)
- When encrypting the first ebs volume,aws kms creates a default cmk key
- This key is used for our first volume encryption.Encryption of snapshots created from this volumes and subsequent volumes created from these snapshots
- After that each newly encrypted volume is encrypted with a unique/seperate AES-256 bit encryption key.
- This key is used to encrypt the volume,its snapshots and any volumes created of the snapshots

Changing Encryption Key :

- We cannot change the encryption(CMK) key used to encrypt an existing encrypted snapshot or encrypt EBS volume
- If we want to change the key,create a copy of the snapshot and specify,during the copy process,that we want to re-encrypt the copy with a diff. key
- This comes in handy when we have a snapshot that was encrypted using our default CMK key and we wnat to change the key in order to be able to share the snapshot with other accounts

Sharing EBS Snapshot :

- By default,only the account owner can create volumes from the account snapshot
- We can share our unencrypted snapshots with the aws community by making them public
- Also we can share our unencrypted snapshot with a selected aws accounts by making them private then selecting aws accounts to share with
- We can not make our encrypted snapshot public
- We cannot make a snapshot of an encrypted ebs volume public on aws

- We can share our encrypted snapshot with specific aws accounts as follows:

□ Make sure that we use a non default/custom cmk key to encrypt the snapshot not the default cmk key(aws will not allow the sharing if default cmk key is used)

□ configure cross account permissions in order to give the accounts with which we want to share the snapshot,access to the custom cmk key used to encrypt the snapshot

□ Without this the other accounts will not be able to copy the snapshot,nor will be able to create volumes of snapshots

- aws will not allow us to share snapshots encrypted using our default cmk key
- For the aws accounts with whom an encrypted snapshot is shared :

□ They must first create their own copies of the snapshot

□ Then they use that copy to restore/create ebs volume

- We can only make a copy of the snapshot when it has been fully saved to S3(its status show as complete) and not during the snapshots pending status(when data blocks are being moved to S3)
- Amazon S3 server side encryption(SSE) protect the snapshot data-in-transit while copying
- We can have upto 5 snapshots copy request running in a single destination per account

LABS :

Session 19 : Creating AMI and recreating EC2 in another region with that AMI

Session 20 : Copy AMI into another AWS Account and recreate EC2

Session 21 : Attach root volume with another EC2 Instance