S3:
Object-storage: No heirarchy
↳ Static Websites & Resources

Only 100 buckets/account
∴ use Service Ticket

Conform to DNS names
and SSL/TLS:

CORS → direct resources in different buckets

completely flat: (key-value store)

Delimiter presents it as a directory structure

S3 Events → λ, SNS, SQS : Added/updated

ingest → free
data trasfer out ⇒ $

11 9's Durability

IA → cheaper: (30day minimum)

(99.9 % object availability
99 % SLA)

one-zone IA:
99.5 %

↗ object retrieva
fee $

Glacier: (do not use for hot backups)

Each version has a new ID:
Without ID → Latest is accessed

→ unique &
isolated: di
tre

Delete the "delete" marker
↳ Previous wala becomes current

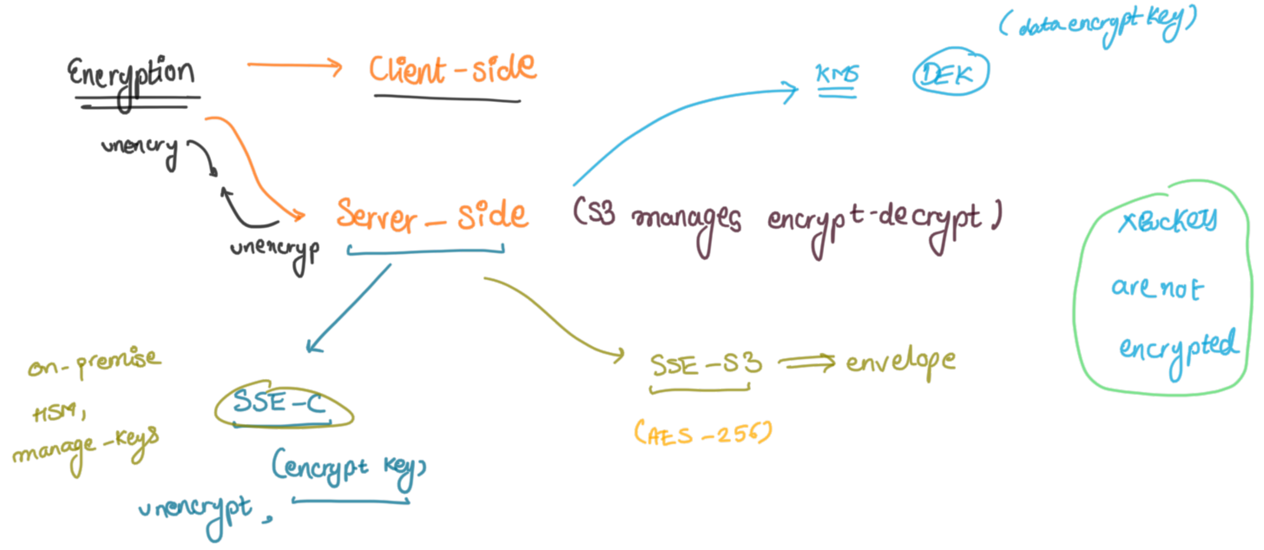
when we create buckets
→ enable locking → can work with versioning only

① Bucket polices ↗ IP Based → Certain time → type of resource encryption
only to S3 Bucket

Pre-signed URL's :⇒ even if object does not exist | Based on rules of user who created the UR

CRR: Replicate ↗ Diff. Region
↳ Can't replicate SSE-C encrypted

Diff. accounts
⇓ Bucket policy



S3 performance: → ① Std v/s Multi-part → ① initiate as MP

(5GB) (Single stream of data)

↳ (n/w vulnerability)

② (upload_ID)

10000 parts 5TB

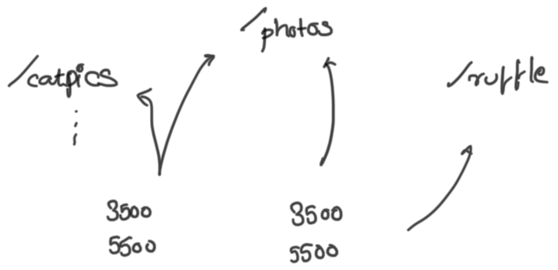
(5M → 5GB)

T/F Accl:

↳ given additional endpoints

Than direct regional endpoint

Partition: '/' Delimiter (3500 PUTS 5500 GETS) / partition



Glacier has vaults and they have archives

No user-defined metadata

can't edit only delete

EFS: → shared filesystem (NFS protocol) / To mount into file system ⇩ EFS / Within a VPC

(in a AZ) shared storage for Linux

↳ 'Mount Target' → each AZ, IP

High throughput and massively scalable

Max I/O mode:

(MKFS) ↳ mount on folder

(Not for temporary storage)

DataSync → from other filesystems

FSx: Managed third-party servers → Windows (SMB) → Lustre (high throughput)

In single AZ → needs Micro AD

VSSBackup

DFS Replicati

Storage G/w: → migration or extension of storage

extend or migrate

(VTL)

backup restorin

File G/w

file

(:1

Volume G/w

extend

block storage

Block storage

Tape G/w ⇓

manage

maintain

obj

S3

SMB, NFS
large file server

migrate

(Not paying for migrating into S3)

(File servers)

Cached → primary S3

Stored G/W → S3 snapshots

iSCSi
replicates to S3, Glacier

store &
Logistics
(backup i
AW

Databases :   ① EC2 Self-Managed Databases : ↗ Root level OS access

Models :   Relations : ↗ ACID

No-Sql : BASE → Key-Value
↓
Eventually consistent

No obligation for consistently
↓
high performance

Key-Value
DynamoDB
Redis

DocumentDB

(Column
↓
Faster for analytical)

Redshift
X transactions

---

① RDS : → inside a subnet in an AZ

Multi-AZ : synchronous replication

(rds) cname

M   S can't access directly

((all data is stored on EBS volumes))

((only encrypt while creating))

Backups : OR (snapshots)
(automatic) ↙ directly
default 7 days
↳ : 0-35 days

Transactional
point-in-time recove

only node not the volume itself ,,

(create a brand new from snapshots)

Read Replica → asynchronous ↗ read heavy

retention period stays even after deleting instance

Aurora : (They share cluster storage volume in same Region)   Primary & Secondary Replicas (Reads)

[illegible] RR
(15 replicas)
but same cluster
Volume

No Need to specify storage   use directly by
using endpoint

Backtrack
↓ w/o changing instance

Aurora clone?

((Backups are incremental and continous

((enable replica Auto Scaling))

Aurora Global DB: Global, very small lag: promote remote in event of failure

Athena:

Serverless, SQL on S3//   XML, CSV ....
Str, semi, un

Schema-on-read

overlay data and look through this schema
(Viewed through the schema
No etl needed: amount of data processed)

No permanent schema
Column - formatted data
⇓
better performance
Does not alter the data
⇒ Process logs), open datasets

Route 53 → public and private
↓
host & register

Alias → logical Aws Resources

in LB → (if one instance fa
can't detect
(individual services

inbound & outbound endpoints: failover: primary secondary

item
(400 KB)

DynamoDB: item↔row
↗
((unique Primary Key)) → only mandatory field of an item) 0 or more attributes

Consuming 1
read/write capa
unit

Sort key will impro

Partition/Hash

Sort Key → 2 elements of primary key

performance

(∵ Query is the Best !)

Indexes: Query based on item value

diff. PK and SK

eventually consistent

Local sec indexes
↓
only when we create

Global :

((Backup & Restore))

DynamoDB user data is encrypted at rest

Elasticache → Key: Value

Advanced:

Mem

- single
- instance
- scales

Redis (lists, hashes, ... adv data struc)

Complex data

Multi-AZ

(Backup & Restore)

Failover auto

Multi-threaded

Map Reduce:

EMR → BigData Frameworks

manages cluster, health monitoring

Master Node

(HDFS name node)

connect here SSH

From S3 → To S3

data rep across all core → Core Node → 0 or More

HDFS data nodes

HDFS storage management

Task → lesser risk of losing

(all cluster components are in same AZ)

Same instance mein master and core node

can't change master node-type

instance fleet ⇒ 5 diff. types of

optimize cost based on s

① Region as close to S3 bucket region

② Recent type of instances

Master is not demanding C

on compute

discrete - piece of application :

Kinesis : Large streams : from large no. of producers

$\underbrace{\text{Name, partition, Data}}_{\text{shard}}$

Multiple consumers

of streams

EC2 : KCL

λ,

FireHose

data records: 1MiB

① Kinesis streams : 24 hour

(ingestion/read req) shards