UVA | SCHOOL of DATA SCIENCE

Distributed Storage

Big Data Systems
School of Data Science
University of Virginia

Agenda

> Objects
> Filesystems
> Amazon S3
> HDFS

What are Objects?

Objects are fundamental units of data storage in cloud computing

Stored in flat namespace (not in a hierarchical structure)

Each object has three parts:
- > data
- > metadata
- > unique identifier (key)

Objects: Additional Properties

> Objects are often immutable.
  Typically overwrite an object rather than making edit.

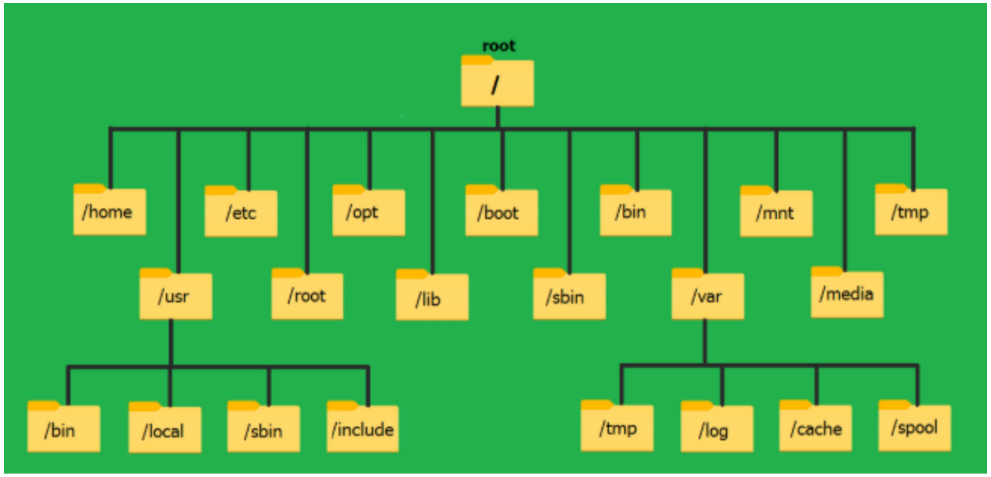
> Since objects are identified by unique keys, they can be spread across multiple servers and data centers

Filesystems

What is a Filesystem?

A filesystem (FS) is the method an operating system uses to organize, store, and retrieve files on a storage device

Here is the Linux FS:



Filesystem Properties

A real filesystem gives you things like:

> Hierarchical directories (that actually exist)

> In-place mutation (write at an offset)

> File locks (to prevent user overwrites and data corruption)

> Cheap metadata operations

Amazon S3 vs. HDFS

Next, we will look at:

> Amazon S3, which uses object storage **for scale**

> HDFS, which uses a distributed filesystem for **throughput**

> They intentionally offer different user contracts
regarding data consistency, performance, and functionality

Amazon S3

Introducing Amazon S3

> Amazon's managed cloud storage solution

> Simple Storage Service; not so simple any more

> Handles durability, replication, scalability behind the scenes

> Object storage built to store and retrieve any amount of data from anywhere

Amazon S3 Core Ideas

> Stores data as *objects* in *buckets*

> Each object has a key (string), metadata, and content

> Effectively infinite scale

> No cluster to manage

Amazon S3 Durability

Amazon S3 Standard and most S3 storage classes designed to provide 99.999999999% durability ("eleven nines") over a year.

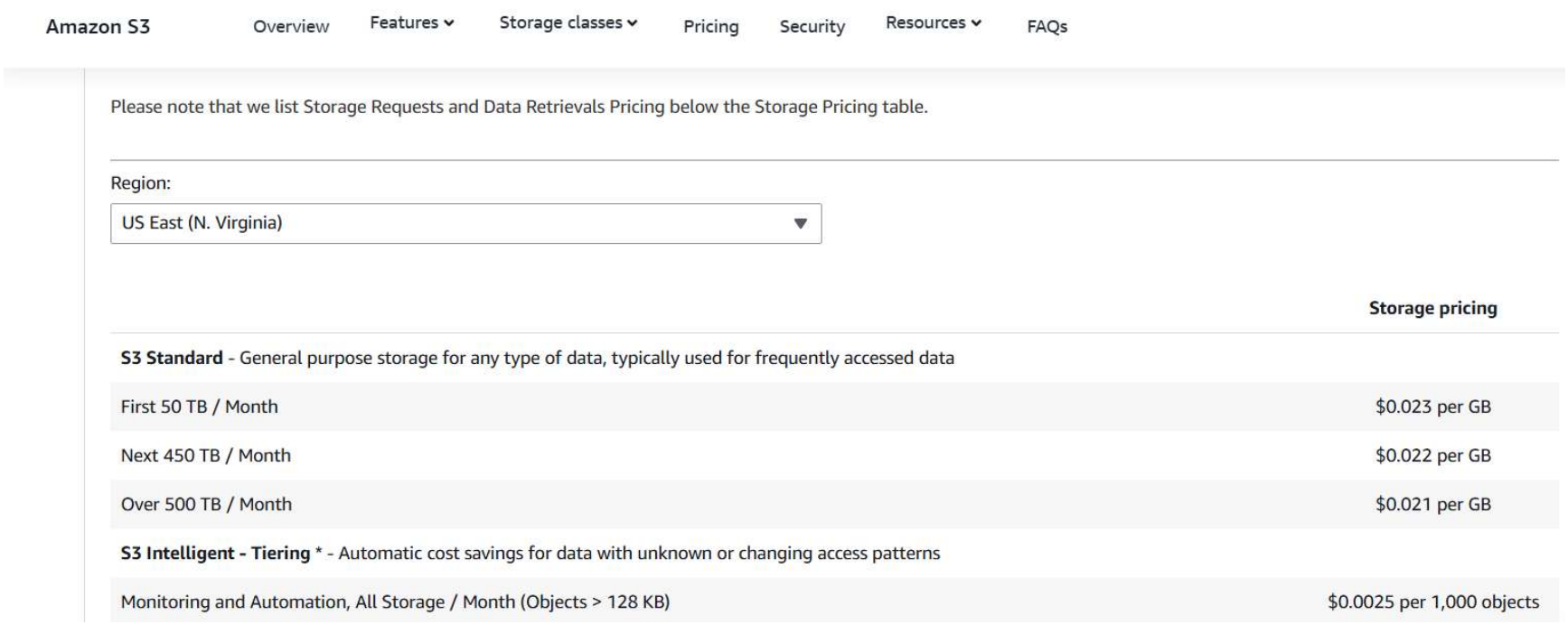
What this means:
For every 10,000 objects stored, you'd expect to lose one object on average every 10 million years

Amazon S3 Replication

> S3 is durable thanks to **replication** across multiple data centers and advanced data protection

> AWS automatically copies data across at least three physically separated Availability Zones within a single region.

Amazon S3 Pricing



https://aws.amazon.com/s3/pricing/

Amazon S3 Limitations

> Higher latency than local disks or HDFS

> No true filesystem (“Directories” are just key prefixes)
- Access is via API calls, not file handles
- Overwrites are whole-object operations
- Can’t do in-place mutation
- No file locks

HDFS

About HDFS

> Hadoop Distributed File System (HDFS) uses a directory structure

> Stores very large files (GB–TB scale) across a cluster of commodity machines

> Built for batch analytics, especially MapReduce / Spark-style workloads

About HDFS

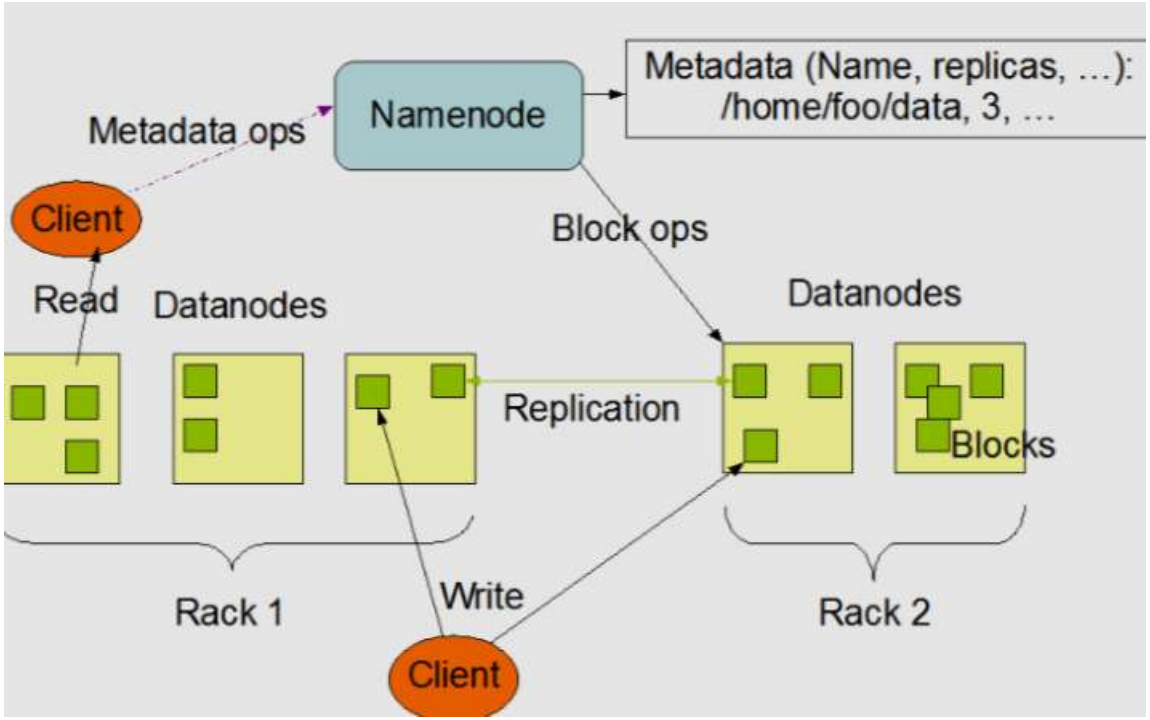
> Write once, read many (append is allowed; random writes are not)

> Files split into large blocks (typically 128–256 MB)

> Each block replicated (3 copies) across different nodes for fault tolerance

> Optimized for high throughput, not low latency

HDFS Architecture



> *NameNode* manages metadata (file paths, block locations, permissions)

> *DataNode* stores blocks of data

> Client asks Namenode where blocks live, then streams data directly from DataNodes

HDFS Strengths

> Excellent for large sequential reads/writes
  - Due to block design

> Strong data locality (compute runs where data lives)
  - NameNode directs compute to the data
  - Replicas provide multiple places to run task

> Tight integration with Hadoop ecosystem (YARN, Spark)

When it Doesn’t Fit Well

> When you don’t have an Ops team to manage

> When you need low latency

> When you have a lot of small files

Where it’s Used

HDFS has been replaced by object storage for many workloads

> Still shows up for jobs running on-prem

> Useful when you need tight data locality

> Common when organization runs a Hadoop cluster

Key Takeaways

> S3 and HDFS were designed for different purposes

> We saw their tradeoffs around scalability and throughput

> Increasingly, object stores are used for distributed storage
  - They can be decoupled from compute