ADALTAS

(Not only) Cloud Storage

Stephan BAUM

Cloud native

ADALTAS

> *Cloud native technologies empower organizations to build and run scalable applications in modern, dynamic environments such as public, private, and hybrid clouds. Containers, service meshes, microservices, immutable infrastructure, and declarative APIs exemplify this approach.*

> *These techniques enable loosely coupled systems that are resilient, manageable, and observable. Combined with robust automation, they allow engineers to make high-impact changes frequently and predictably with minimal toil.*

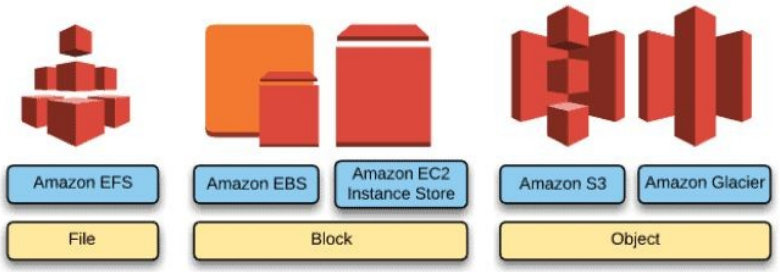
- Not per-se related to cloud, but to application architecture

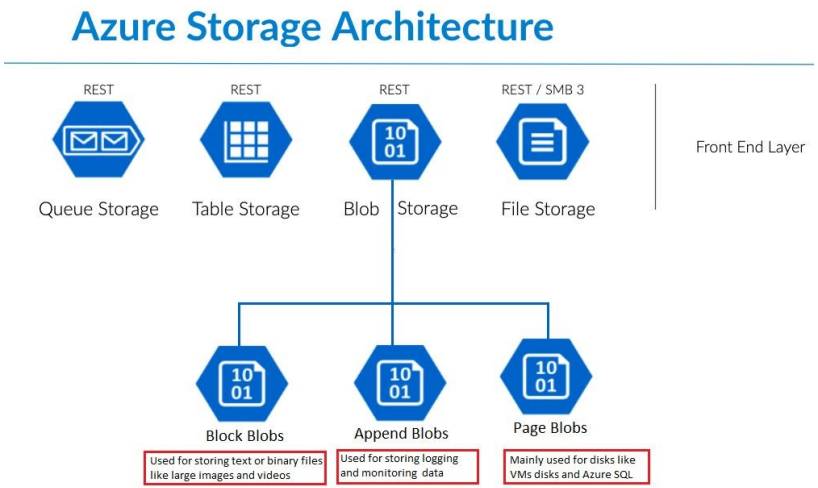
Cloud storage formats

ADALTAS

	Object	Block	File
Price	$	$$$	$$
Data Type	Semi and Unstructured Data	Structured Data	Structured, Semi Structured and Unstructured
Scale	Massive Scale (starts small from TBs and grow to EBs seamlessly	Very few TBs	TBs to PBs
Protocols	S3 API	Block Protocols (iSCSI)	CIFS, NFS
Workloads	AI/ML, Big Data, Streaming, Database, Snapshots, Backup	Transactional DB	Archive, User generated files
Approach	Object = File + Metadata + Globally unique identifier	File is written on block on spinning media	File is stored and limited metadata is stored along as a separate entry

Cloud storage formats





https://www.youtube.com/watch?v=hKWWGvC5_uo

Cloud native storage

ADALTAS

- Requirements:
  - S3 compatible API
  - built for Kubernetes
- Cloud native storage is **object storage**
  - can handle huge amounts of data
  - affordable
  - sufficiently fast for the applications
  - distributed
  - resilient
  - highly available
- Storage and compute are **decoupled** (we can scale them independently)

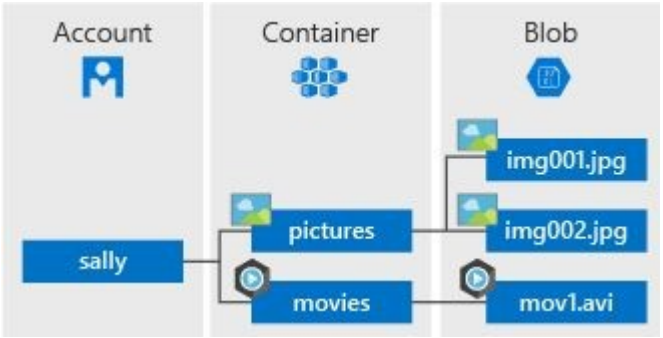
What is an object?

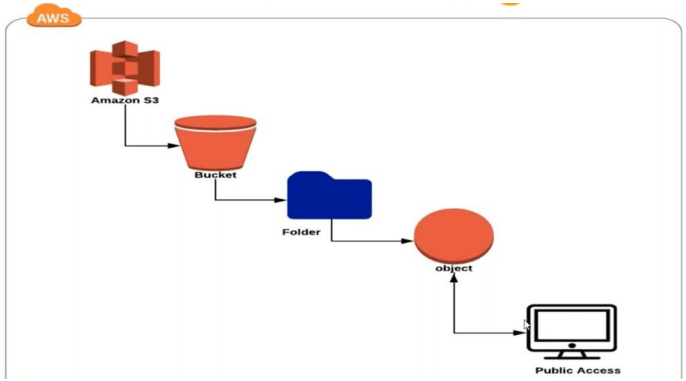
ADALTAS

- **Components:**
  - data:
    - structured data: database snapshots
    - unstructured: videos, audio files
    - semi-structured: logs
    - max size for individual object (5 TB S3 bucket object, 200 TB Azure blob storage)
    - binary objects (not the same as blocks)
  - network address:
    - http://s3amazonaws.com/<bucket-name>/<object-name> (for publicly available)
  - metadata:
    - user that created it, last access…
    - checksum
    - lifecycle policy (expiration date)

How an object is stored?

- Objects are saved as **key-value pairs**
  - key: network address
  - value: the data
- It is **immutable**

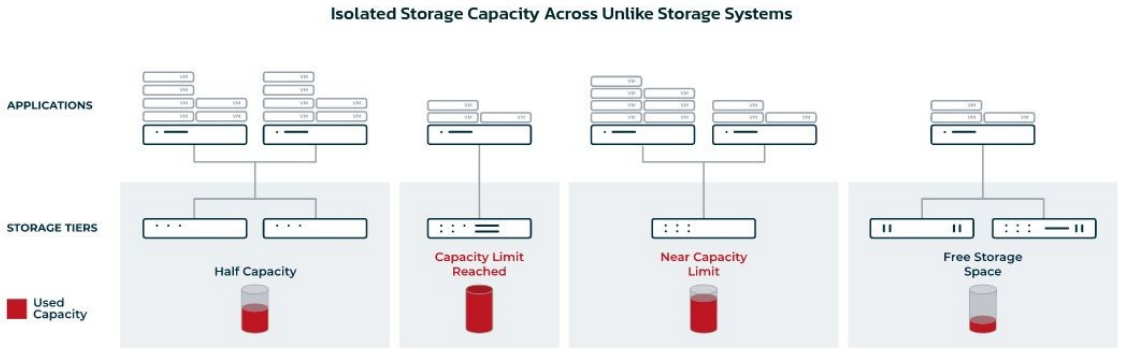




Virtualized storage, Software defined storage

ADALTAS

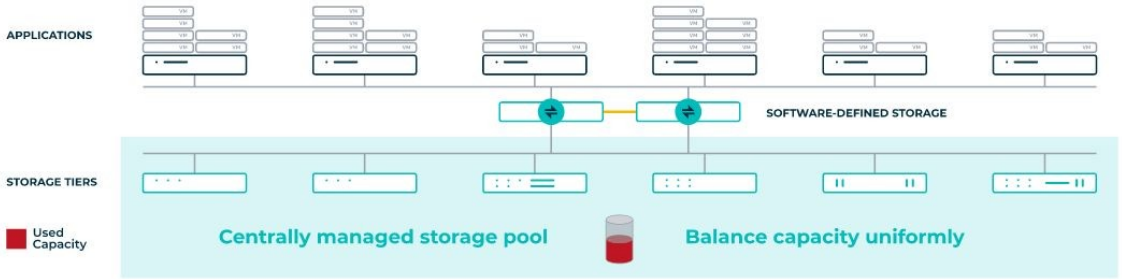
Before SDS



Without Software-Defined Storage

- Monolithic and siloed storage architecture
- Rigid, hardware-bound storage infrastructure
- Can run into vendor lock-in
- Storage refresh and data migration are tedious
- Premium devices are stressed with overload
- Add more hardware when low on capacity

After SDS



With Software-Defined Storage

- Centrally pooled and fluid storage architecture
- Flexible and hardware-independent storage infrastructure (hardware is decoupled from storage software)
- Ultimate freedom to choose any storage vendor/model/type
- Non-disruptive storage refresh and data migration
- Balance capacity and load uniformly across diverse and unlike storage systems
- Optimize unused capacity intelligently and defer new hardware expenditure

Virtualized storage, Software defined storage

ADALTAS

- **Virtualized** storage:
  - decoupling the hardware and capacity
  - we can join different storage devices into one big storage pool
  - the resources will be shared among users/applications

- **Software-defined** storage:
  - separates the hardware from functional aspects of the storage
    - security
    - identity and access management
    - predicate pushdown
    - data fault-tolerance
  - could implement objects, blocks and files
  - proprietary and open-source projects

MinIO: Software defined storage solution

ADALTAS

- Cloud-native object storage
- Highly scalable
- Erasure coding and RotBit protection
- Lifecycle management
- ...