Using MINIO object storage for digital preservation tasks

Jonáš Svatoš, Head of Digital Laboratory
Národní filmový archiv, Prague

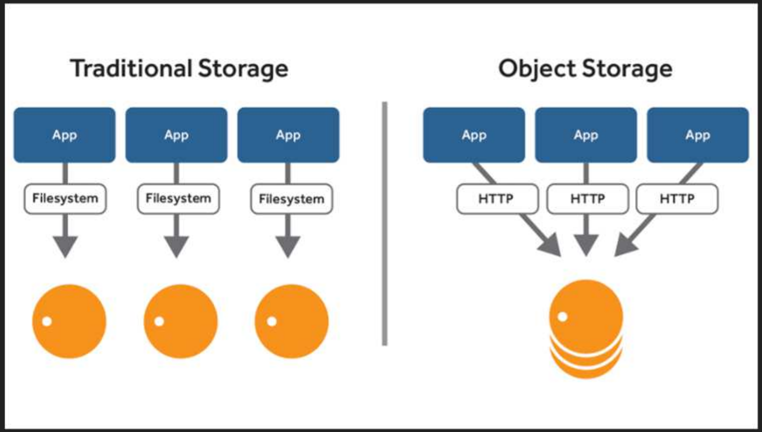
No Time To Wait! #4
Budapest, 6.12.2019

Motivation

- Traditional file systems do not scale well for AV
- SAN is complicated (and $$$)
- Redundancy or Performance? Pick one
- Microservices do not like filesystems

What’s an “object storage” anyway?

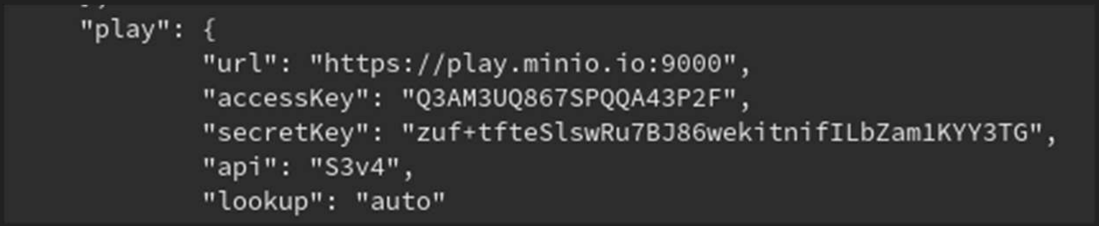
- Data as objects, instead of files
- Object is UUID + data + metadata
- Web APIs as a storage abstraction (usually REST API)
- Top-level folder = Bucket
- “Folders” are just metadata



Source: doc.aws.amazon.com

A word about security

- No more “security by obscurity”
- Whole data storage is accessible via REST API
- Access control via secrets present in every HTTP/S request
- Tighter access-control requirements



```
"play": {
        "url": "https://play.minio.io:9000",
        "accessKey": "Q3AM3UQ867SPQQA43P2F",
        "secretKey": "zuf+tfteSlswRu7BJ86wekitnifILbZam1KYY3TG",
        "api": "S3v4",
        "lookup": "auto"
```

Národní filmový archiv

Multiple implementations, one API

Hosted

- Amazon S3
- Google Cloud Storage
- ..

On-premise

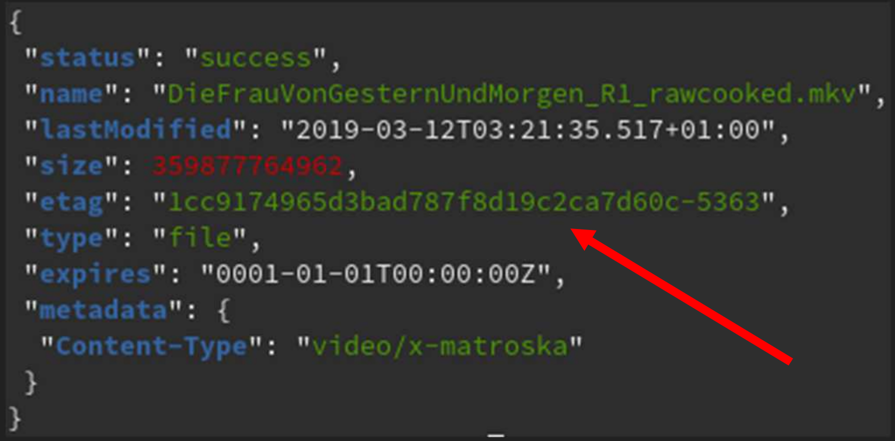
- Ceph
- MinIO
- OpenIO

MinIO

- “Do one thing, and do it well”
- Written in Go, one binary
- Data chunking as a way towards parallelization
- Multi-GB/s speeds on commodity hardware w/spinning disks
- Inherent redundancy and bit-rot protection
- Both standalone and cluster-aware

Fixity

- S3 API enforces checksum calculation and retention
- Data chunking speeds things up by calculating hashes in parallel
- MD5 hash function (so-2000's)
- Hash in every HTTP response
- https://github.com/antespi/s3md5



```
{
 "status": "success",
 "name": "DieFrauVonGesternUndMorgen_R1_rawcooked.mkv",
 "lastModified": "2019-03-12T03:21:35.517+01:00",
 "size": 359877764962,
 "etag": "1cc9174965d3bad787f8d19c2ca7d60c-5363",
 "type": "file",
 "expires": "0001-01-01T00:00:00Z",
 "metadata": {
  "Content-Type": "video/x-matroska"
 }
}
```

Redundancy and bit-rot protection

- MinIO supersedes RAID by employing Erasure coding
  - $ minio server /mnt/disk{1..32}
- Configurable redundancy (N/2 + 1 by default)
  - Even when half of drives +1, still able to write to it
- Uses HighwayHash internally (up to 10GB/s on single core)
- Automatic bit-rot detection and correction

WORM mode

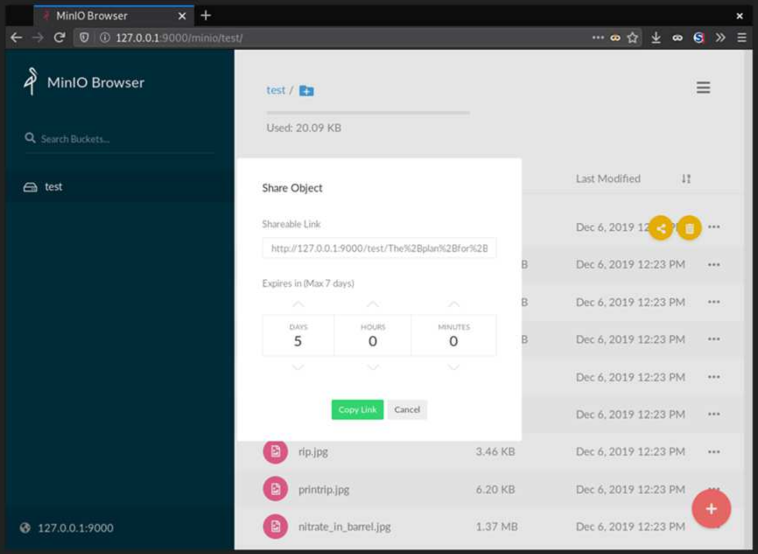
- Write once, read many
- Only read and write, no delete/move/overwrite

Classical Filesystem interface

- For some workflows, filesystem interface is required
- S3 has a wrapper implementation (FUSE, mostly POSIX-compliant)
- Retains parallelization benefits
- Metadata access is expensive though (no DPX sequences please..)
- https://github.com/s3fs-fuse/s3fs-fuse

Tools

- Native Web UI
- CLI client - mc
- S3cmd
- Cyberduck
- ..

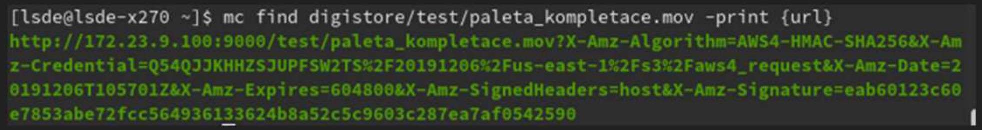


Shareable Link: http://127.0.0.1:9000/test/The%2Bplan%2Bfor%2B

Expires in (Max 7 days)

DAYS	HOURS	MINUTES
5	0	0

		Dec 6, 2019 12
	B	Dec 6, 2019 12:23 PM
	B	Dec 6, 2019 12:23 PM
	B	Dec 6, 2019 12:23 PM
		Dec 6, 2019 12:23 PM
		Dec 6, 2019 12:23 PM
rip.jpg	3.46 KB	Dec 6, 2019 12:23 PM
printrip.jpg	6.20 KB	Dec 6, 2019 12:23 PM
nitrate_in_barrel.jpg	1.37 MB	Dec 6, 2019 12:23 PM



```
[lsde@lsde-x270 ~]$ mc find digistore/test/paleta_kompletace.mov -print {url}
http://172.23.9.100:9000/test/paleta_kompletace.mov?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Am
z-Credential=Q54QJJKHHZSJUPFSW2TS%2F20191206%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=2
0191206T105701Z&X-Amz-Expires=604800&X-Amz-SignedHeaders=host&X-Amz-Signature=eab60123c60
e7853abe72fcc564936133624b8a52c5c9603c287ea7af0542590
```

Links

- https://github.com/minio/minio
- https://docs.aws.amazon.com/AmazonS3/latest/API/s3-api.pdf
- https://github.com/google/highwayhash
- https://github.com/s3fs-fuse/s3fs-fuse
- https://github.com/antespi/s3md5

Thank you

jonas.svatos@nfa.cz
github.com/NFAcz