A MAINFRAMER'S GUIDE TO OBJECT STORAGE

Object storage is coming to the mainframe. It's the optimal platform for demanding backup, archive, disaster recovery (DR), and big-data analytics operations, allowing mainframe data centers to leverage scalable, cost-effective cloud infrastructures.

For mainframe personnel, object storage is a new language to speak. It's not complex, just a few new buzzwords to learn. This paper was written to introduce you to object storage, and to assist in learning the relevant terminology. Each term is compared to familiar mainframe concepts. Let's go!

What is Object Storage?

Object storage is a computer data architecture in which data is stored in object form – as compared to direct-access storage devices (DASD), file/NAS (network-attached storage) storage and block storage. Object storage is a cost-effective technology that makes data easily accessible for large-scale operations, such as backup, archive, DR, and big-data analytics and business intelligence (BI) applications. IT departments with mainframes can use object storage to modernize their mainframe ecosystems and reduce dependence on expensive, proprietary hardware, such as tape systems and virtual tape libraries (VTLs).

Basic terms

Let's take a look at some basic object storage terminology (and compare it to mainframe lingo):

- **Objects**. Object storage contains *objects*, which are also known as *blobs*. These are analogous to mainframe *data sets*.
- **Buckets**. A *bucket* is a container that hosts zero or more objects. In the mainframe realm, data

sets are hosted on a *volume* – such as a tape or DASD device.

Data sets vs. objects – a closer look

As with data sets, objects contain both data and some basic metadata describing the object's properties, such as creation date and object size. Here is a table with a detailed comparison between data set and object attributes:

NOTE: The object attributes described below are presented as defined in Amazon Web Services (AWS) S3 storage systems.

Attribute	Mainframe Data Sets	Object Storage
File system structure	Non-hierarchical, no directory structure. Governed by catalogs and VTOCs.	Non-hierarchical, no directory structure. Delimiter characters can be used to emulate a directory structure.
Size	Limited by the size of the physical media (tape, DASD) used and the DSNTYPE of the data set.	Maximum object size: 5 TB
Scalability	Storage capacity is limited by the size of the physical media (tape, DASD) used. The number of data sets is also limited by VTOC, VVDS and catalog size. Larger sizes cause degradation in performance.	"Directories" are virtual, and can span multiple hardware devices. Therefore there are no physical or performance limits on the number of objects stored.
Naming	Data set names are all caps, limited to 44 characters. Example object name: REPORTS.D200320.T223000.PRDPLX1.PS	Object names are case sensitive, limited to 1024 characters. The object name is also known as the *object key*. Example object name: reports/2020-03-20/22:30:00-prdplx1.txt
Name delimiters	'.' (dot) is used to delimit name qualifiers	Any character can be used as a name qualifier. Usually, '/' is used to simulate the Unix directory structure
Metadata	Standard metadata – VTOC / VVDS / Catalog has to be carefully managed to be kept in sync among different systems	Standard metadata, plus: Custom metadata in key → value format, example: DocumentAuthor → "John Doe"
Indexing	Data sets are indexed in VTOCs and catalogs, which are managed by the MF storage administrator.	Indexing is not managed by the storage administrator.
Listing	Data sets are sorted/filtered by prefix.	Objects are sorted/filtered by prefix.
Modifiability	The content of data sets can be modified multiple times.	Objects are not modifiable, they can be created, replaced, and deleted. *Versioning* is a feature used to maintain older versions and deleted versions of the object.
Mainframe Communications	Data sets can be accessed *natively* by z/OS using the proprietary Ficon protocol.	Objects can be accessed using the Open S3 protocol over TCP/IP. z/OS requires an additional software layer to access the object storage

Volumes vs. buckets – a closer look

Buckets, which are analogous to mainframe volumes, are unlimited in size. Separate buckets are often deployed for security reasons, and not because of performance limitations. A bucket can be assigned a life cycle policy that includes automatic tiering, data protection, replication, and

automatic at-rest encryption.

NOTE: The bucket attributes described below are presented as defined in AWS S3 storage systems.

Attribute	Mainframe Volume	Object Storage Bucket
Name	Up to 6 characters.	Up to 63 characters. On cloud platforms, in order to avoid collisions, it is common to add the organization name as a prefix.
Capacity	Limited by the size of the physical media volume (tape, DASD) used. The limitation is partially overcome using *storage groups*. A volume has a Volume Table Of Contents (VTOC) and a VTOC Index that is managed by the storage administrator. The size of the VTOC determines the number of data sets that can be created on the volume.	Buckets can span multiple hardware devices, and can be thought of as an unlimited version of the mainframe storage group. • Bucket size is unlimited. • Buckets can contain an unlimited number of objects.
Scale	The number of volumes is limited by a maximum number of devices – 64K (including secondary and flashcopy devices).	An unlimited number of buckets can be used.
Tiering	• On-premise tapes • DASD HDD device types and RPMs	• Multiple cloud-storage accessibility levels also known as *storage classes*

Security considerations

In the z/OS domain, a system authorization facility (SAF) username and password are required, as well as the necessary authorization level for the volume and data set. For example, users with ALTER access to a data set can perform any action – read/write/create/delete.

In object storage, users are defined in the storage system. Each user is granted access to specific buckets, prefixes, objects, and separate permissions are defined for each action, for example:

- PutObject
- DeleteObject
- ListBucket
- DeleteObjectVersion

In addition, each user can be associated with a programmatic API key and API secret in order to access the bucket and the object storage via a TCP/IP-based API. When accessing data in the cloud, HTTPS is used to encrypt the in-transit stream. When accessing data on-premises, HTTP can be used to avoid encryption overhead. If required, the object storage platform can be configured to perform data-at-rest encryption.

Disaster recovery considerations

While traditional mainframe storage platforms such as tape and DASD rely on full storage replication, object storage supports both replication and erasure coding. Erasure coding provides significant savings in storage space, as the data can be spread over multiple geographical locations. For example, on AWS, data is automatically spread across a minimum of 3 geographical locations, thus providing multi-site redundancy and disaster recovery from anywhere in the world. Erasure-coded buckets can also be fully replicated to another region, as is practiced with traditional storage.

Most object storage platforms support both synchronous and asynchronous replication.

Connecting object storage to the mainframe

BMC AMI Cloud Data is a software-only platform that leverages powerful, scalable cloud-based object storage capabilities for data centers that operate mainframes.

The platform runs on the mainframe's zIIP processors, providing cost-efficient storage, backup, archive, and recovery functionalities with an easy-to-use interface that requires no object-storage knowledge or skills.