IBM Cloud Object Storage System

WekaFS Filesystem

IBM COS Solution Guide
Version 1.1

Table of Contents

Revision History

Version	Date	Description
1.0	6/1/2020	
1.1	6/23/2020	

Integration information

AIT Certification Engineer	Jim Kladis
Publication Date	
AIT Review	6/19/2020
Target ClevOS Version	3.14+
Target Partner Product Version	3.6.0/3.7.0

Introduction

The purpose of this solution guide is to describe the **IBM Cloud Object Storage**'s AIT certification of **Weka** clustered filesystem, and tiering to **IBM COS**.

Functional and performance iterations were carried out on the **Weka** nodes (shared client) and remote clients over NFS, CIFS and the WekaFS agent.

Intended Audience

This guide is intended to assist solution architects, integrators, and end users.

Scope

The guide sections describe the components as they pertain to the solution, the configuration elements used in an integration, and results of the arranged tests.

Terminologies

- **Agent**: The client software ensuring the client is on the correct version of the cluster.
- **Backend:** One or most **Weka** hosts operating within the cluster.
- **Client:** Software for the client to connect to the Weka backend hosts and be part of a cluster.
- **Cluster:** Logical unit of backend hosts that makes up the scalable, distributed filesystem.
- **Drive Capacity:** The configured amount of SSD capacity allotted to a filesystem.
- **Tiering Cue/Demote:** The configured period of time for data to be tiered from local disk to object storage.
- **(Filesystem) Group:** Logical grouping of filesystems that share a data retention configuration.
- **Converged Deployment**: A shared client model, where the user application server and the **Weka** backend are on the same system.
- **Prefetch**: The **Weka** process of rehydrating data from an object store to an SSD, based on a prediction of future data access.
- **Retention**: The maximum period of time for data to be retained on local disk after being tiered to object storage. Data is released (deleted) from SSD after this time.
- **DPDK**: Data Plane Development Kit, the preferred protocol for client access to the WekaFS, is a set of libraries and network drivers for highly efficient, low latency packet processing.
- **Fetch**: Rehydrating data back to disk from object storage.
- **Failure Domain**: A group of **Weka** backends that can encounter an outage while maintaining data protection.
- **OBS**: Object Store (contraction).
- **Protection Level**: The protection level is the number of additional protection blocks (concurrent failures) added to each stripe, which can be either 2 or 4.
- **Provisioned Capacity**: The configured amount of total capacity which includes the object store.

- **Stripe Width**: The stripe width is the number of blocks that share a common protection set, which can range from 3 to 16 (the smallest possible stripe width is 3+2).
- **GUID**: The ID assigned to an entire Weka cluster. The GUID, its prefix, and suffix are used in the S3 prefix of some object workflows.
- **Serial Number**: An arbitrary object/blob identifier that increases within a single bucket raft-term/incarnation used in the S3 prefix of some object workflows.
- **Snap-to-Object (S2O):** The feature of uploading FS snapshots directly to object storage.
- **Snap Layer Generation**: A number that is stored in the cluster configuration for each snapshot, and is increased when the snap is modified by merge operations due to older snap deletions, is used in the S3 prefix of some object workflows.
- **Read Cache:** File data is consistent across hosts and stored on disk. *There may be some metadata inconsistency in extreme cases.*
- **Write Cache:** Default state utilizing a single Linux host page cache**.** In this state, the highest performance is possible, once data is consistent.

Solution Introduction

The **Weka** matrix is a scalable, high-performing parallel file system. The system operates as cluster of “backend” nodes running the **Weka** software which furnishes a POSIX-compliant filesystem. This filesystem gets accessed by application clients by either an installed driver for maximum performance, or by commonly known SMB/CIFS and NFS file sharing protocols.

SSD drives for storing data are installed on these servers and accessed directly outside of the system drivers. The resultant storage system is scalable to hundreds of backends and thousands of clients with the lowest possible latency between the drives and the application access.

The solution offers a data lifecycle capability so that older data can be copied to **IBM COS** and later be released from SSD to free up space. Filesystem level snapshots can be taken and immediately uploaded to object storage. Data managed by the Weka system is typically considered either hot (resident on the SSD) or warm (readily available on object storage), as opposed to a cold, long archive.

Solution Scope

The certification solution tests the **WekaFS** filesystem protocol operations (e.g. NFS GETATTR, READ, WRITE, REMOVE, etc.) and its ability to manage data that has been tiered to the cloud.

Feature	Solution Support	Certification Functional Testing	Certification Performance Testing
Data Lifecycle Tiering	YES	YES	YES
Data Fetch	YES	YES	YES
File Protocol Access	YES	YES	YES
Space Reclamation	YES	YES	NO

Solution Deployment

Deployments model tested.

1. Functional validation.
    a. **Weka** system was tested at its filesystem and sharing protocol levels.
    b. The data lifecycle policy of tiering data to object storage, retrieval and space reclamation.
    c. **IBM COS** System (on-premise)
        i. Vault mode with packed storage.
        ii. Container mode with zone storage.
    d. **IBM COS** Public.

2. Performance analysis.
    a. Observe filesystem data and metadata performance.
    b. Measure performance with **IBM COS** System (on-premise).

The solution certification executed and documented scenarios to a local on-premise **IBM COS** and then **IBM COS** Public over the public internet.

Note: The public internet connection is limited to 200 Mbps. A public internet OBS is supported but not ideal for lower latency fetches.

Solution Limitations

- **Weka** system version 3.7.0 introduces space reclamation (deletes). Earlier versions do not routinely delete from the configured object store.
- For an object store status to change from “Up” to “Down”, all nodes must lose connectivity to **IBM COS**.

Note: A configured group’s lifecycle policy (retention) of files is calculated based on the filesystem’s attaching to object storage + the group policy tiering/retention periods.

Performance Limitations

- Mounted (external) clients for certification testing was performed in UDP mode, over TCP/IP Ethernet. The published expected maximum client performance over UDP mode is approximately 2 GB/s of throughput.
- Both **IBM COS** and **Weka** system are scale out and scale up solutions. See the test architecture for configurations used.
- The network link between the AIT lab and the **Weka** system cluster is 100 Gbps.
- Throughput performance to/from the **IBM COS** Public service carried some observed variations.
    - Depending on upstream and downstream used, speeds seen range from 200 Mbps to 1 Gbps in some cases.
    - When possible, averages of multiple runs are featured.

CSO Interface Usage

S3 Client

The Weka system uses the Boto3 S3 Python SDK to perform operations to and from **IBM COS** and supports AWSv2 and v4 signatures (configurable).

The User Agent string seen is: `APN/1.0 WEKA/1.0 WEKASCALE/V3.6.0.0`

Security (Encryption In-Transit)

TLS 1.2 is supported with **IBM COS System** (on-premise) and **IBM COS** Pubic (cloud). During TLS handshake, the Boto3 client offers numerous TLS cipher suite combinations, including strong, and also older, obsolete ones.

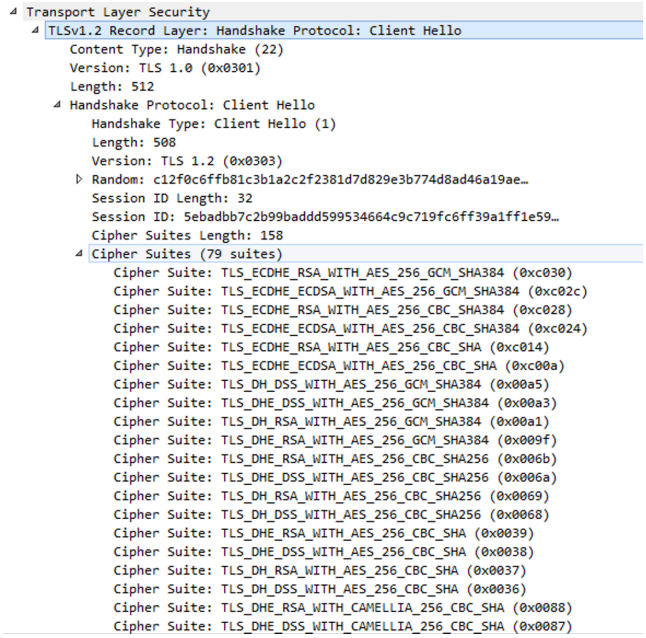


Figure 1. Weka system S3 Client Hello TLS cipher suite offer list (trimmed) capture.

IBM COS System and **Weka** negotiated TLS 1.2 under the `TLS_RSA_WITH_AES_256_CBC_SHA256` cipher suite.

IBM COS Public and **Weka** negotiated TLS 1.2 under the `TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384` cipher suite.

AWS Key/IAM

The Weka system accepts key/secret format used on **IBM COS** System and **IBM COS** Public with IAM +HMAC.

Threading/Parallelism/Concurrency

Every backend node in the **Weka** cluster participates in communicating with **IBM COS** in a distributed fashion. S3 operations are initiated by every healthy host under normal circumstances. The configurable cluster-wide concurrent upload or downloads is set to a maximum default of 64 connections.

Object, ACLs, and Bucket Naming

The **Weka** system can be configured to address up to 2048 buckets. It does not automatically create or otherwise manage buckets, so Read/Write (**IBM COS** System) and `serviceRole:Writer` (**IBM COS** Public) are sufficient permissions for the **Weka** system operations.

When first configured to an existing **IBM COS** bucket, the **Weka** system will complete a verification series of essential operations by performing a `PUT` of a 1 byte object, a `GET` retrieval, and finally attempting a `DELETE`.

Object Type	Prefix	Example
Data Blob	`B_<ReversedBucketID>/<GUID>/d/I_<ReversedSerialNumber>/<RaftTerm>`	`B_321/00000000-0000-0000-0000-000000000000/d/I_987/456`
Metadata Blob	`B_<ReversedBucketID>/<GUID>/m/I_<ReversedSerialNumber>/<RaftTerm>`	`B_321/00000000-0000-0000-0000-000000000000/m/I_987/456`
Manifest Chunk Blob (Snapshot Data)	`B_<ReversedBucketID>/<GUID>/m/I_<ReversedChunkIndex>/<SnapLayerId>/<SnapLayerGeneration>`	`B_321/00000000-0000-0000-0000-000000000000/m/I_654/345/567`
Spec Blob	`<GUID Prefix>/d/s/<SnapLayerId>/spec/<GUID Suffix>`	`00000000/d/s/345/spec/0000-0000-0000-000000000000`
Bucket Validation (PUT then GET)	`<GUID Prefix>/verify/<Node ID>/rw/<GUID Suffix>`	`00000000/verify/678/rw/0000-0000-0000-000000000000`
Bucket Validation (Attempt DELETE of non-existent object)	`<GUID Prefix>/verify/<Node ID>/remove/<GUID Suffix>`	`00000000/verify/678/remove/0000-0000-0000-000000000000`

Table 1. Weka object prefix/naming structure.

The same bucket name can be used across multiple **Weka** Object Store configurations. In addition, the same Object Store configuration can be applied to multiple filesystems.

For organizational purposes, the bucket path can be an appended prefix (e.g. `bucketname/top-level-object-name`).

> *Bucket name parsing in **Weka** is case-sensitive. Configure the bucket name using the same case that is displayed in **IBM COS**.*

Data that's moved from **Weka** to **IBM COS** is packed into 64 MiB (default maximum) objects (blobs). The contents of a data blob object are concatenated ranges of the Weka FS 4 KB blocks.

Operation Mapping

During normal IO, a standard `PUT` is used to upload an object. A `GET` of either the entire object or partial object (range read) are used for retrieval. In **Weka** system 3.7, space reclamation will perform a standard `DELETE` of objects as space is reclaimed.

Use case	CSO Operations Mapping
Check object storage bucket	REST.GET.OBJECT
Copying data to object storage	REST.PUT.OBJECT
Fetching data from cloud	REST.GET.OBJECT (with or without `bytes=`)
Delete objects for space reclamation	REST.DELETE.OBJECT

Table 2. Weka system operation types.

Test Architecture

Testing was conducted from the AIT lab environment to four COS types.

On Premise, Single Site, High Performance

Weka Cluster

- **Supermicro BigTwin 2 Server Solution**
  - 8 Storage Nodes
  - 16 Intel Xeon Gold 6126 2.6GHz CPU
  - 768 GB DDR4
  - 184.32 TB of Micron 9300 Pro U.2 SSD
    - 5 node Striping + 2 Data Protection Blocks with 1 Hot Spare
    - 104 TB Usable Capacity
  - Mellanox ConnectX-4 100Gbit/s VPI Dual Port PCIe
    - 100G Ethernet Ports Were Used
  - Cisco Nexus Non-blocking 100G Ethernet Switch for Backend Cluster Connectivity and Uplink to COS with Jumbo Frames Enabled
- Converged Mode (DPDK) Client Testing for Full Client Performance
- Jumbo Frames Enabled

IBM COS Gen 1

- IBM COS version 3.14 – Vault and Container Mode
- 4 4105 Accessers
  - 40 Gbps links
- 12 2448 Slicestors
  - 48 Seagate 6TB 7200 RPM SATA Drives
  - 40 Gbps links
- Client to Accesser latency is local
- 12 Wide Storage Pool
- Vault Configuration
  - 12/7/9 IDA
  - SecureSlice: Disabled
  - Name Index: Enabled
  - Segment Size: 4 MiB

- Single Accesser, DNS Round Robin, and Reverse Proxy Load Balanced S3 Endpoints using Citrix Netscaler (40G)
- Jumbo Frames Enabled

AIT Clients

- 3 Ubuntu 18.04 Linux Clients - VMWare
  - 100 GB Local SSD
  - 100 GB Ethernet – VMXNET3
  - WekaFS Client – UDP Mode
  - NFSv3 Client

IBM COS Public Cloud

- General Purpose Shared WAN Internet Connection
  - Approximately 40ms RTT to COS Public Endpoint `s3.us.cloud-object-storage.appdomain.cloud`

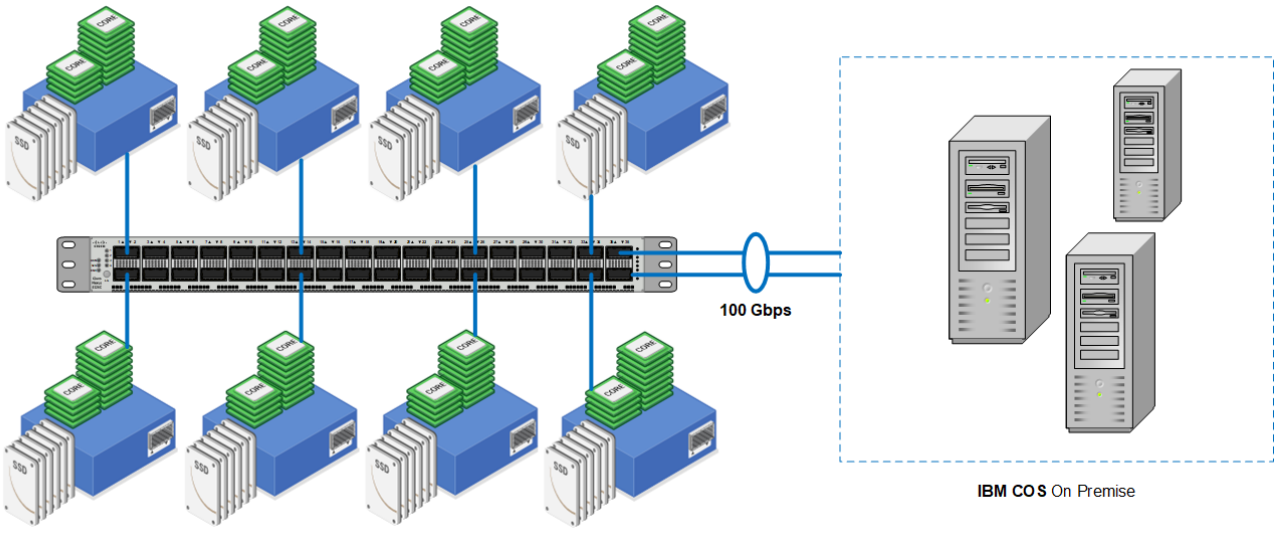


Figure 2. On-Premise cluster overview.

Functional Characterization

The **Weka** system accepts file writes and distributes them to the cluster accordingly. To a client mounted to a WekaFS, the filesystem appears as a standard POSIX-compliant mounted FS, with free disk space reported as the allocated total capacity.

In a filesystem with an attached object storage, data that meets the retention criteria is promptly tiered in an orderly fashion, with oldest files being tiered first.

Tiering

In tiered **Weka** system scenarios, there are various locations for data storage:

- Active metadata is stored only on SSD.
- Snapshot metadata is stored on SSD and copied to object storage.
- Writing of new files, adding data to existing files or modifying the content of files is always committed to the SSD, regardless of whether the file is already stored on the SSD or tiered to object storage.
- When reading the content of a file, data can be accessed from either the SSD (if it is available on the SSD) or rehydrated from the object store (if it is not available on the SSD).

Fetching

Data will be retrieved (fetched) under the following conditions:

- A tiered file's contents are read by an application.
- A snapshot that resides on object storage is restored to an existing or as a new filesystem.
- An object store is detached from a filesystem, causing all tiered data to be rehydrated.

Deleting

Data can be deleted (space reclamation) based on the following automated design:

- The **Weka** system will attempt to allow for 7%-13% more object store usage than required. For each filesystem that exceeds the 13% threshold, the **Weka** system will only re-upload objects for which logically more than 5% of them are freed (and will gather those objects in a full 64 MB object again).
- The **Weka** system will stop the reclamation process if the filesystem consumes less than 7% of its object store space, to avoid high writes amplifications and allow some time for higher portions of the 64 BM objects to become logically free. This ensures that the object storage will not be overloaded when just reclaiming small portions of space.

While the steady state of a filesystem requires up to 13% more raw capacity in the object store, this percentage may increase when there is a load on the object store (which takes precedence) and when there is frequent deletion of data/snapshots. Over time, it will return to the normal threshold after the load/burst is reduced.

The **Weka** system reclaims object store space at optimal capacity level, but will reduce to background priority to ensure filesystem operations are executed first.

Encryption

When using an encrypted filesystem, data is encrypted by the client/shared client as it's written to SSD and decrypted as it's rehydrated.

Performance Characterization

Migration of warm data and snapshots to object storage are executed in the background and is designed to not interfere with **Weka** system's primary function of serving IOs with high performance.

Performance measurements are based on the following types of operations.

1. File operations using converged, WekaFS, SMB, and NFS clients.
2. Data tiered to **IBM COS**.
3. Data recalled from **IBM COS** and rehydrated to SSD.
4. Inline recall, where data is fetched from COS and served to the client, while simultaneously rehydrated to SSD.

See Test Results for findings of performance and testing.

WekaFS encryption was tested and is seen to affect client performance but was not seen to affect cloud tier/fetch performance.

Performance Scaling

Weka Node Count

The **Weka** system has the ability to effectively utilize multiple **IBM COS** Accessers. It automatically distributes the S3 requests across the N nodes of the **Weka** cluster to the configured S3 endpoint, relative to where the SSD data is stored.

Weka Client Count

Clients utilizing resources for .

Object (Blob) Sizing

In most cases, the default maximum object blob size of 64 MiB provided the greatest balance of performance and backend CPU efficiency. The tuning of object sizes smaller than 64 MiB are possible. Smaller objects tend to use more backend CPU, though may be more appropriate for some situations.

With HTTPS transport and COS On-Premise, 64 MiB causes TCP window receive buffer overrun, and was addressed with configuring smaller blobs.

The minimum configurable size is 4 MiB.

Reverse Proxy Load Balancing

The Citrix Netscaler 40G load balancer was tested and max TCP throughput saturation achieved.

DNS Round-Robin Load Distribution

The majority of performance runs were achieved by deploying an on-premise DNS hostname that returned multiple Accesser IP addresses. In the current hardware configuration, 4 Accessers was adequate for best HTTP performance and Accesser availability, and allowing resources available for other IO.

For HTTPS, CPU usage was significantly higher, and 8 Accessers were used.

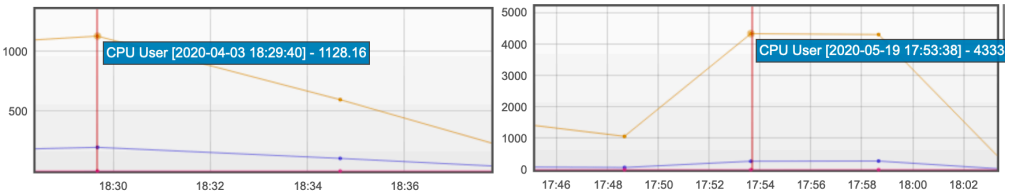


Figure 3. **IBM COS** 4105 Accesser GET CPU impacts, HTTP (Left) vs. HTTPS (Right).

Example (DIG) DNS Lookup Response:

```
;; ANSWER SECTION:
4accpod3.ait.cleversafe.com. 3600 INA      172.20.33.11
4accpod3.ait.cleversafe.com. 3600 INA      172.20.33.14
4accpod3.ait.cleversafe.com. 3600 INA      172.20.33.13
4accpod3.ait.cleversafe.com. 3600 INA      172.20.33.12
```

Capacity Scaling

Both **IBM COS** and the **Weka** cluster are designed to scale limitlessly, with practical constraints existing mainly at the available hardware and licensing levels.

The **Weka** system is licensed by both drive capacity and object store capacity. The licensing entitlement is what determines the maximum configurable total capacity.

When capacity planning a Weka FS, a key consideration is provisioned capacity, which is the total capacity assigned to a filesystem. The sum includes initial SSD *and* overall object store capacity.

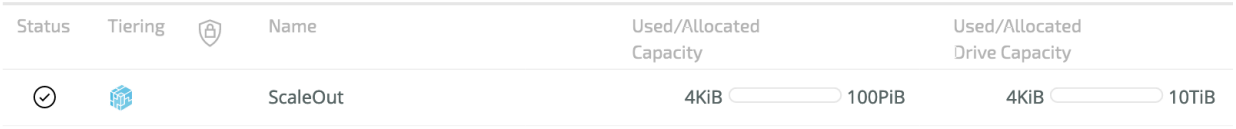


Figure 4. Example filesystem with 10 **TiB** of drive capacity and 100 **PiB** of provisioned capacity.

If the **Weka** system cannot comply with the defined Retention Period, e.g., the SSD is full and data has not been released from the SSD to the object store, a Break In Policy will occur. In such a situation, an event is received in the **Weka** system event log, advising that the system has not succeeded in complying with the policy and that data has been automatically released from the SSD to the object store, before completion of the defined Retention Period. No data will be lost (since the data has been transferred to the object store), but slower performance may be experienced.

The observed amount of constant SSD storage for metadata stored on the **Weka** system was observed to be around 5-15% of the total test data generated. The percentage varies depending on use case.

> *If the drive capacity used for metadata exceeds the allocated capacity, then drive expansion must take place.*

If a **IBM COS** System hard quota is exceeded, a HTTP response status 507 will be sent, preventing uploading until the exceeded storage is remedied.

Test Results

Baseline Throughput Performance.

Baseline throughput performance is achieved either by threading Goader tests, snapshot upload/download (after workload was generated) or object store detach (download).

Single IBM COS 4105 Accesser

Protocol	Tier	Throughput	Fetch	Throughput
HTTP	8 TB/HR[1]	2.18 GB/s	4 TB/HR[2]	1.32 GB/s
HTTPS	5 TB/HR[1]	1.33 GB/s	3.5 TB/HR[1]	980 MB/s

Table 3. Single 4105 endpoint with Genset Jumbo file workload.

Reverse Proxy Load Balancer

Protocol	Tier	Throughput	Fetch	Throughput
HTTP	14 TB/HR	3.85 GB/s	16 TB/HR	4.25 GB/s
HTTPS	11 TB/HR	3.11 GB/s	12.5 TB/HR	3.48 GB/s

Table 4. Citrix NetScaler VPX (40000) with 4 Accesser service group, and Genset Jumbo file workload.

DNS Round-Robin

Protocol	Tier Data Rate	Throughput	Fetch Data Rate	Throughput
HTTP	31 TB/HR	8.62 GB/s	26 TB/HR	8.62 GB/s
HTTPS	13 TB/HR	2.37 GB/s	21 TB/HR	5.7 GB/s

Table 5. 4 Accesser DNS A-record, with Goader 100 MB file workload.

Source File Size Types

File Sizes	Tier Data Rate	Throughput	Fetch Data Rate	Throughput
100 MB	31 TB/HR	8.62 GB/s	26 TB/HR	8.62 GB/s
1 MB	26 TB/HR	7.24 GB/s	16 TB/HR	4.42 GB/s
128 KB	13 TB/HR	7.85 GB/s	6 TB/HR	1.62 GB/s

[1] Bound by Accesser CPU utilization.
[2] Bound by Accesser NIC receive from SliceStor nodes.

Table 6. 4 Accesser DNS A record, with Goader various file workload, over HTTP.

Blob (Object) Size Tuning - HTTP

Blob Size	Tier	Throughput	Fetch	Throughput
16 MiB	33 TB/HR	9.17 GB/s	26 TB/HR	6.25 GB/s
32 MiB	31 TB/HR	8.62 GB/s	21 TB/HR	5.94 GB/s
64 MiB (Default)	31 TB/HR	8.62 GB/s	26 TB/HR	6.25 GB/s

Table 7. HTTP 4 Accesser DNS A record, with Goader 100 MB file workload, with various object sizes.

Blob (Object) Size Tuning – HTTPS

When using the default blob size, the upload throughput was negatively impacted by TCP Window/Buffer window updating. As this may be related to sharing backend NICs with connectivity to object storage, it has been observed to be mitigated with smaller blob (object) sizes.

Blob Size	Tier	Throughput	Fetch	Throughput
8 MiB	27 TB/HR	7.44 GB/s	33 TB/HR	9.31 GB/s
16 MiB	26 TB/HR	7.2 GB/s	28 TB/HR	7.85 GB/s
32 MiB	22 TB/HR	6.23 GB/s	28 TB/HR	7.72 GB/s
64 MiB (Default)	13 TB/HR	2.37 GB/s	21 TB/HR	5.7 GB/s

Table 8. HTTPS 8 Accesser DNS A record, with Goader 100 MB file workload, with various object sizes.

Use Case Analysis

A sampling of tests was performed on a converged client (run directly on Weka nodes) that compares the timing of first executing with the source file only on object storage, then following “fetching” is back on read cache (SSD) and executed again.

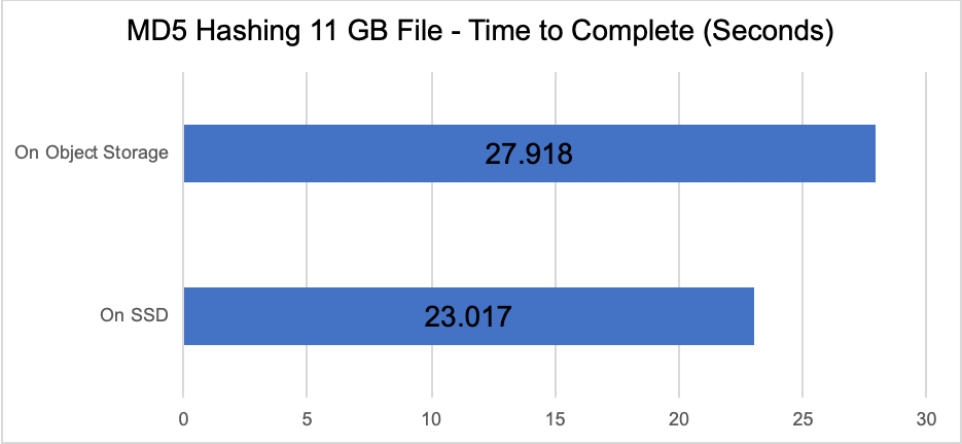


Figure 4. Time comparison to complete an “openssl md5” operation of an 11 GB file.

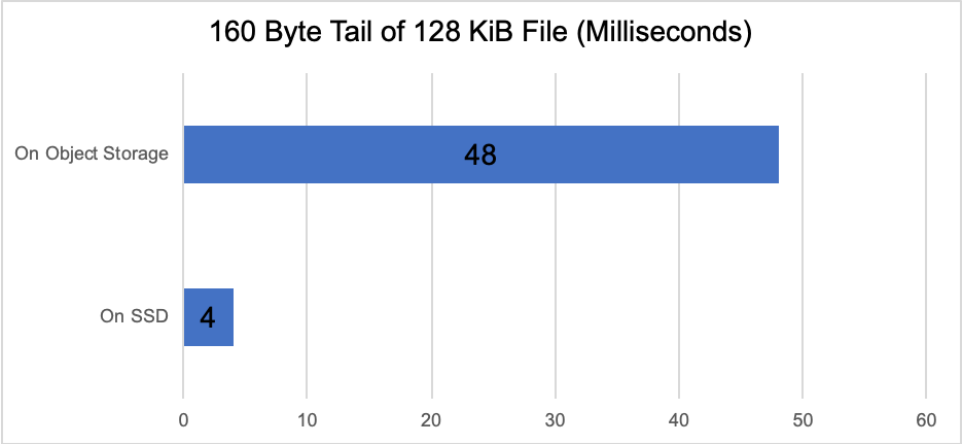


Figure 5. Time comparison to read 160 bytes of a 128 KiB file.

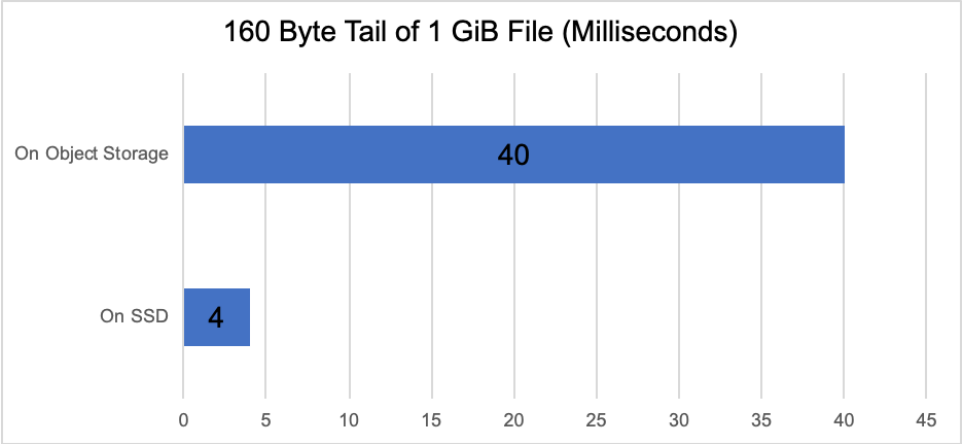


Figure 6. Time comparison to read 160 bytes of a 1 GiB file.

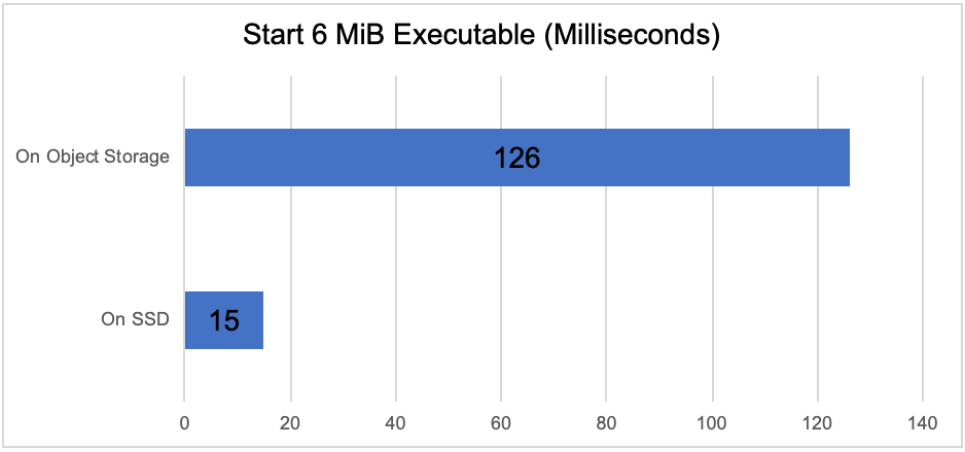


Figure 7. Time comparison to start a 6 MiB executable file.

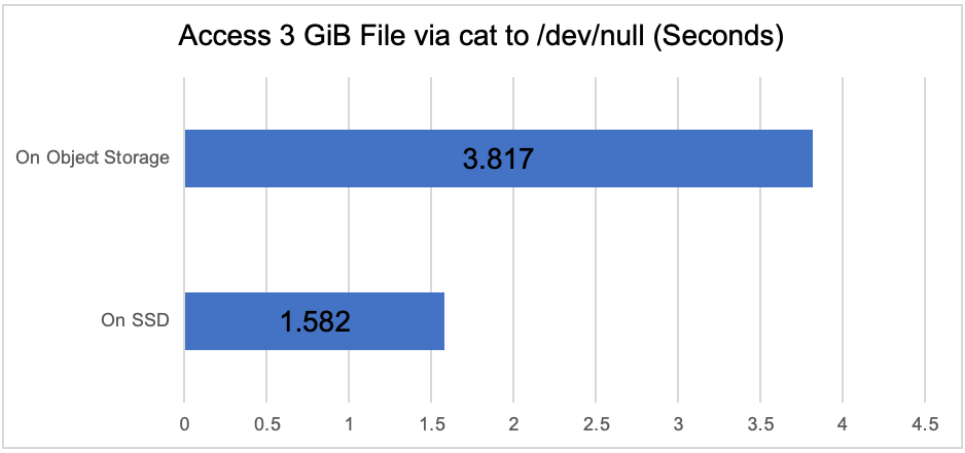


Figure 8. Time comparison to read a full 3 GiB file (via cat to /dev/null).

IBM COS Operations Analysis

Tiering to Cloud

- 1.4 TB of tiered data. Goader 100 MB file workload.
- Default Object Size.
- **IBM COS** On Premise.

IO Type	Request Method	Op-Class	Count	HTTP Status	Number of Accessers	Latency Avg.	Object Size Avg
Write	REST.PUT.OBJECT	A	21,875	200	4 (DNS)	645 ms	64 MB

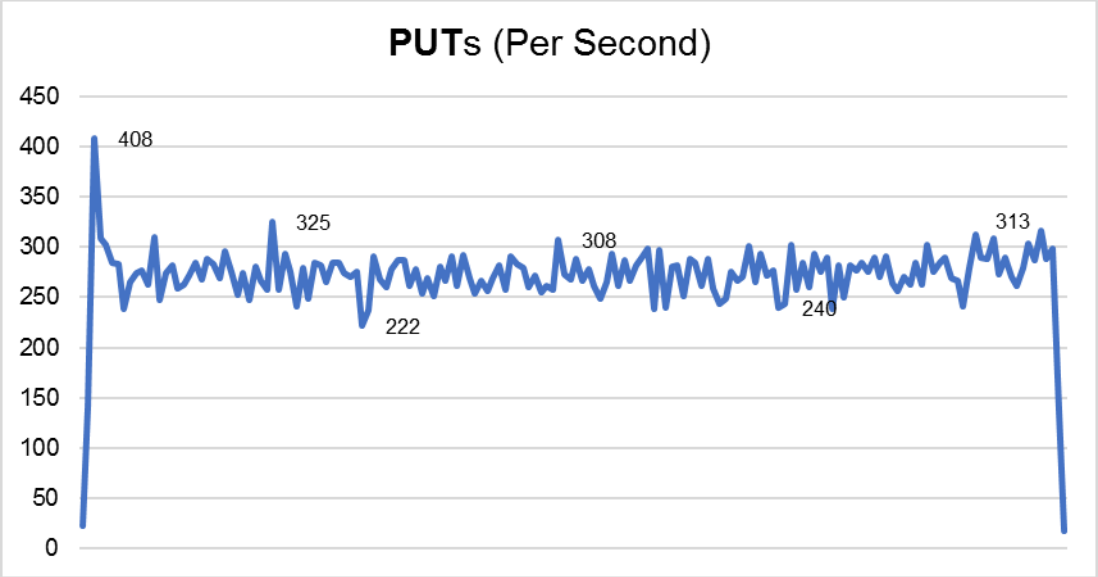


Figure 9. **IBM COS** upload PUT operations.

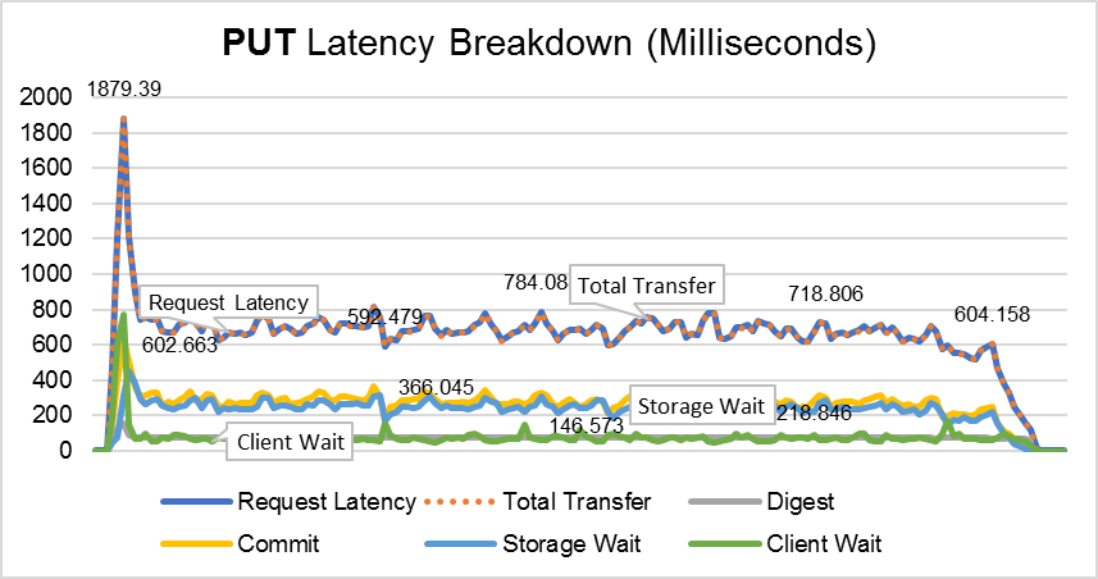


Figure 10. **IBM COS** upload PUT latencies.

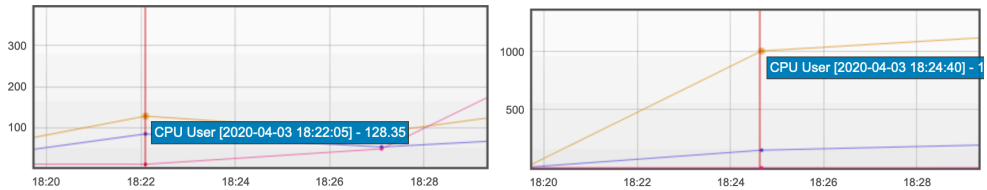


Figure 11. **IBM COS** SliceStor (Left) and Accesser (Right) PUT CPU impacts.

Fetch from Cloud

- 1.4 TB of tiered data. Goader 100 MB file workload.
- **IBM COS** On Premise.

IO Type	Request Method	Op-Class	Count	HTTP Status	Number of Accessers	Latency Avg.	Object Size Avg
Read	REST.GET.OBJECT	B	2,160,000	206	4 (DNS)	112ms	64MB (1 MB Range)

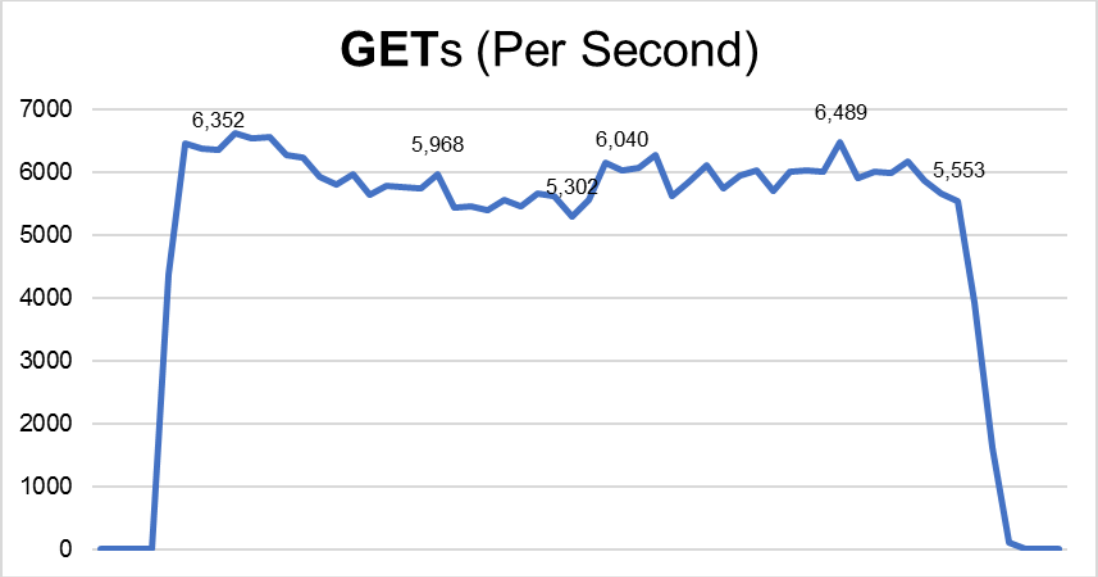


Figure 12. **IBM COS** download GET operations.

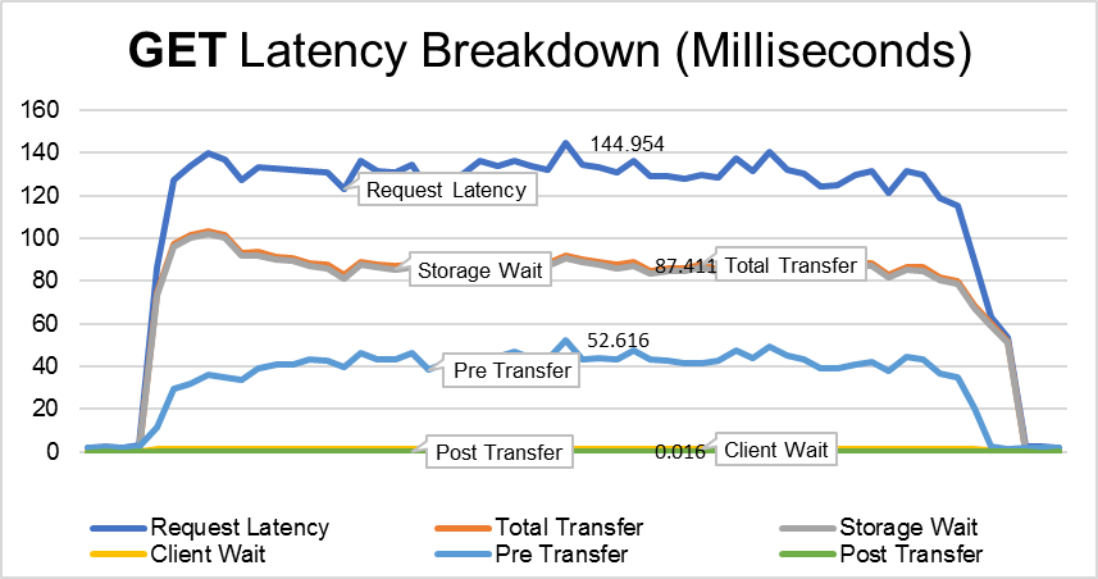


Figure 13. **IBM COS** download GET latencies.

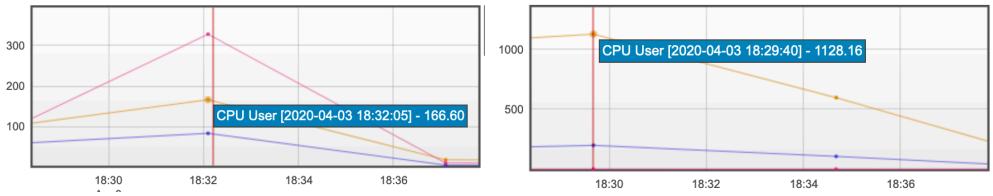


Figure 14. **IBM COS** SliceStor (Left) and Accesser (Right) GET CPU impacts.

Deleting from Cloud

- 7 TB Data.
- **IBM COS** On Premise.

IO Type	Request Method	Op-Class	Count	HTTP Status	Number of Accessers	Latency Avg.	Object Size Avg
Delete	REST.OBJECT.DELETE	N/C	108,000	204	4 (DNS)	375ms	64MB

Persistent Operations

Approximately every second, a single (1) `REST.GET.OBJECT` keepalive operation to the configured **IBM COS** endpoint is sent from a random node in the cluster.

The response is a 63 byte keepalive object, totaling in a daily egress bandwidth of ~5.4 MiB.

Tools

- **Weka** system statistics used for data rate and cross-referenced by **IBM COS** and network tools.
- **Genset** for mixed IO workload and use case testing.
    - https://github.ibm.com/cs-team-ait/genset
- **Goader** for FIO-based static file and threaded testing.
    - https://github.com/tigrawap/goader

Appendix A: Configuration Guide

Preparation

1. Select an existing or create a new **Weka** filesystem.
2. Confirm your tiering and retention design.

DATA IN A FILESYSTEM MAY BE A CANDIDATE FOR TIERING ONCE AN OBJECT STORE IS ATTACHED.

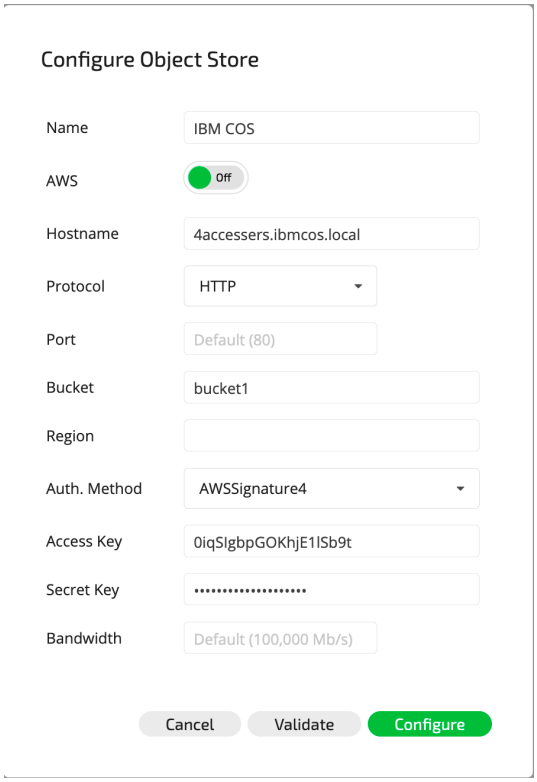
---

3. Confirm your load balancing strategy and design.
    a. If using DNS load balancing, create the host (A) records for the appropriate number of Accessers.
    b. If using a reverse proxy load balancer, configure the backend/server group for the appropriate number of Accessers.
4. Create a bucket/vault on the COS environment and record the case-sensitive name.
5. Record the following information:
    a. The COS endpoint URL.
    b. The Access Key ID associated with the COS account.
    c. The Secret Access Key associated with the COS account.
6. If the **Weka** system will connect to an on premise **IBM COS** deployment over HTTPS, obtain the proper CA certificate from the load balancer, or **IBM COS Manager** CA for the device(s) that will be used as the COS endpoint.
7. Confirm the user account used to access the **Weka** system GUI possess either the *ClusterAdmin* or *OrgAdmin* roles.
    a. If using CLI, confirm the user account has SUDO/root access to the **Weka** nodes.

Create an Object Store (GUI)

1. Connect to the **Weka** system GUI using a login with either the *ClusterAdmin* or *OrgAdmin* role.
2. Toggle the side menu by clicking on the three line button located in the upper left corner of the page.

3. Under the *Configure* section, select *Object Stores*.
4. From the main object store view screen, click the "+" button located in the upper left corner of the page. The Configure Object Store dialog box will be displayed.



5. Configure the parameters.
    a. **Name**: Friendly name for the Object Store.
    b. **AWS**: Toggle off to revert to non-AWS S3 prepopulated configuration.
    c. **Hostname**: FQDN or IP address of Accesser/load balancer/DNS group.
    d. **Protocol**: HTTP, HTTPS or HTTPS_UNVERIFIED (trust Accesser node's manager-signed certificate).
    e. **Port**: Accesser TCP listening port. Only specify if not using the default.
    f. **Bucket**: Bucket name on **IBM COS**, entered case-sensitive.
    g. **Region**: Unused for **IBM COS**.
    h. **Auth. Method**: AWSSignature4 (Recommended) or AWSSignature2.
    i. **Access Key**: HMAC credentials key.
    j. **Secret Key**: HMAC credentials secret.
    k. **Bandwidth (Optional)**: Limit the backend CPU core's upload bandwidth in Mbps.
6. Optionally select *Validate* and finally *Configure*.

IF RECONFIGURING THE OBJECT STORE FROM THE GUI, RE-SPECIFY THE AWS SIGNING METHOD, OTHERWISE A FAILURE MESSAGE WILL BE DISPLAYED.

Create an Object Store (CLI)

1. Connect to a Weka console over SSH using a root/SUDO login.
2. Execute the `weka fs tier s3 add` command.
    a. `name`: Friendly name for the Object Store.
    b. `--hostname`: FQDN or IP address of Accesser/load balancer/DNS group.
    c. `--protocol`: HTTP (Default), HTTPS or HTTPS_UNVERIFIED (trust Accesser's manager-signed certificate).
    d. `--port` **(Optional)**: Accesser TCP listening port. Only specify if not using the default.
    e. `--bucket`: Bucket name on **IBM COS**, entered case-sensitive.

f. `--region`: Unused for **IBM COS**.
g. `--auth-method`: AWSSignature4 (Recommended) or AWSSignature2.
h. `--access-key-id`: HMAC credentials key.
i. `--secret-key`: HMAC credentials secret.
j. `--bandwidth` **(Optional)**: Limit the backend CPU core's upload bandwidth in Mbps.

Example:

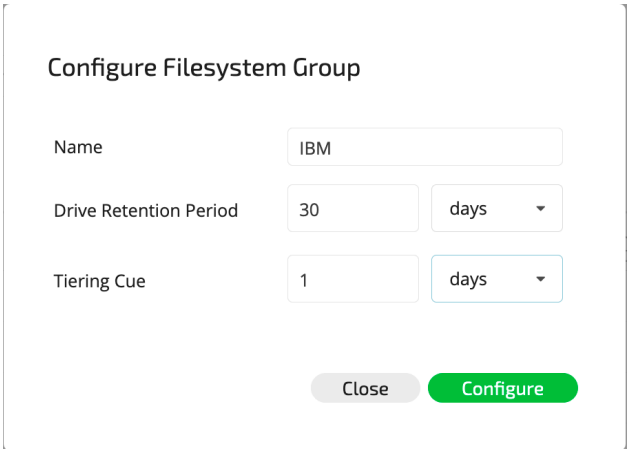
```
weka fs tier s3 add "IBM COS" --hostname 4accessers.ibmcos.local --bucket bucket1 --auth-method
AWSSignature4 --access-key-id 0iqSIgbpGOKhjE1lSb9t --secret-key
JvaPHumyC9jyG8tDQUnkqD6lf5mXss3iC940gdQr --protocol HTTP
```

Configuring the Filesystem Group Retention (GUI)

1. Connect to the **Weka** system GUI using a login with either the *ClusterAdmin* or *OrgAdmin* role.
2. Toggle the side menu by clicking on the three line button located in the upper left corner of the page.

3. Under the *Configure* section, select *Filesystems*.
4. From the main Groups column, select the associated FS Group.
5. Select the *Edit* button.
   a. Configure the *Drive Retention Period* (Delete/Release from SSD after tiering).
   b. Configure the *Tiering Cue.*

Tiering cue must be at least 10 seconds and no larger than (Retention / 3).



6. Select *Configure* to complete.

Configuring the Filesystem Group Retention (CLI)

1. Connect to a **Weka** console over SSH using a root/SUDO login.
2. Execute the `weka fs update` command.
   a. `--target-ssd-retention`: Period of time to keep an SSD copy of the data (format: 3s, 2h, 4m, 1d, 1d5h, 1w)
   b. `--start-demote`: Period of time to wait before copying data to the Object Storage (format: 3s, 2h, 4m, 1d, 1d5h, 1w)

Example:

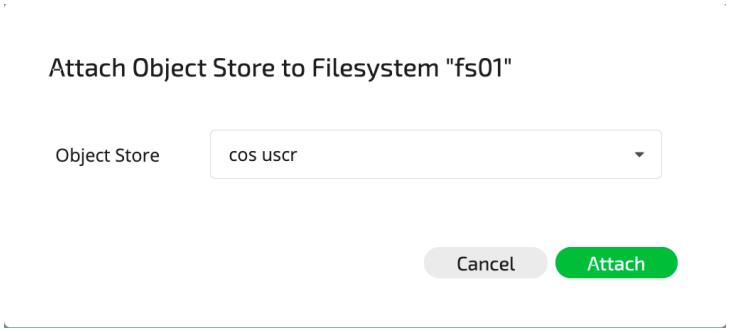
```
weka fs group update IBM --target-ssd-retention=30d --start-demote=1d
```

Attaching an Object Store (GUI)

1. Connect to the **Weka** system GUI using a login with either the *ClusterAdmin* or *OrgAdmin* role.
2. Toggle the side menu by clicking on the three line button located in the upper left corner of the page.

3. Under the *Configure* section, select *Filesystems*.
4. Locate the filesystem to be configured.
5. Select the *Attach Object Store* button.

6. Use the dropdown to select the appropriate object store.
7. Select *Attach* to complete.



Attaching an Object Store (CLI)

1. Connect to a **Weka** console over SSH using a root/SUDO login.
2. Execute the `weka fs tier s3 attach` command.
    a. `weka fs tier s3 attach <fs-name> <obs-name>`

Example:

```
weka fs tier s3 attach fs01 'cos uscr'
```

Verifying Object Store Status

1. Connect to a **Weka** console over SSH using a root/SUDO login.
2. Execute the `weka fs tier` command to see if the object store is considered “UP” or “DOWN”.
    a. `weka fs tier`

Verifying Ongoing Tiering Status

1. Connect to a **Weka** console over SSH using a root/SUDO login.
2. Execute the `weka fs tier ops` command to see operations “in flight”.
    a. `weka fs tier ops <obs-name>`
    b. Optionally `grep Ongoing` for active uploads.

Examples:

```
weka fs tier ops 'cos uscr'
```

```
Node | Object Store Identifier                          | Operation Type | Execution Time | Phase  | Last Execution Time | Start Time           | Size     | Previous Attempt Results | Curl Errors | Concurrency Level | Inode Id
-----+--------------------------------------------------+----------------+----------------+--------+---------------------+----------------------+----------+--------------------------+-------------+-------------------+---------
2    | B_972/a9fc039f-e928-4892-9f2a-114e5d6fcd9b/d/I_3012/491 | Upload         | 0.349165s      | Recent | 0.349117s           | 2020-Jun-10 15:10:45 | 16846848 | SUCCESS: 1               |             | 1                 | 0
2    | B_541/a9fc039f-e928-4892-9f2a-114e5d6fcd9b/d/I_7012/485 | Upload         | 0.429202s      | Recent | 0.429147s           | 2020-Jun-10 15:10:45 | 16846848 | SUCCESS: 1               |             | 2                 | 0
2    | B_618/a9fc039f-e928-4892-9f2a-114e5d6fcd9b/d/I_2602/504 | Upload         | 0.282001s      | Recent | 0.281949s           | 2020-Jun-10 15:10:59 | 8425472  | SUCCESS: 1               |             | 3                 | 0
2    | B_546/a9fc039f-e928-4892-9f2a-114e5d6fcd9b/d/I_8412/513 | Upload         | 0.333581s      | Recent | 0.333554s           | 2020-Jun-10 15:10:59 | 8425472  | SUCCESS: 1               |             | 12                | 0
2    | B_297/a9fc039f-e928-4892-9f2a-114e5d6fcd9b/d/I_6212/510 | Upload         | 0.339029s      | Recent | 0.339001s           | 2020-Jun-10 15:10:59 | 8425472  | SUCCESS: 1               |             | 13                | 0
2    | B_693/a9fc039f-e928-4892-9f2a-114e5d6fcd9b/d/I_6902/493 | Upload         | 0.341971s      | Recent | 0.341926s           | 2020-Jun-10 15:10:59 | 8425472  | SUCCESS: 1               |             | 9                 | 0
2    | B_48/a9fc039f-e928-4892-9f2a-114e5d6fcd9b/d/I_4802/479  | Upload         | 0.345959s      | Recent | 0.345914s           | 2020-Jun-10 15:10:59 | 8425472  | SUCCESS: 1               |             | 4                 | 0
2    | B_822/a9fc039f-e928-4892-9f2a-114e5d6fcd9b/d/I_0712/483 | Upload         | 0.349569s      | Recent | 0.349517s           | 2020-Jun-10 15:10:59 | 8425472  | SUCCESS: 1               |             | 1                 | 0
2    | B_118/a9fc039f-e928-4892-9f2a-114e5d6fcd9b/d/I_9712/487 | Upload         | 0.351272s      | Recent | 0.351235s           | 2020-Jun-10 15:10:59 | 8425472  | SUCCESS: 1               |             | 5                 | 0
```

```
weka fs tier ops 'cos uscr' | grep Ongoing
82 | B_894/a9fc039f-e928-4892-9f2a-114e5d6fcd9b/d/I_9412/524 | Upload | 0.000000s | Ongoing | 0.000000s | 2020-Jun-10 15:19:20 | 8425472 |  |  | 5 | 0
82 | B_374/a9fc039f-e928-4892-9f2a-114e5d6fcd9b/d/I_8312/516 | Upload | 0.000000s | Ongoing | 0.000000s | 2020-Jun-10 15:19:20 | 8425472 |  |  | 3 | 0
82 | B_007/a9fc039f-e928-4892-9f2a-114e5d6fcd9b/d/I_1222/493 | Upload | 0.000000s | Ongoing | 0.000000s | 2020-Jun-10 15:19:20 | 8425472 |  |  | 2 | 0
82 | B_312/a9fc039f-e928-4892-9f2a-114e5d6fcd9b/d/I_7602/520 | Upload | 0.000000s | Ongoing | 0.000000s | 2020-Jun-10 15:19:20 | 8425472 |  |  | 1 | 0
83 | B_028/a9fc039f-e928-4892-9f2a-114e5d6fcd9b/d/I_1712/410 | Upload | 0.000000s | Ongoing | 0.000000s | 2020-Jun-10 15:19:20 | 8425472 |  |  | 1 | 0
83 | B_403/a9fc039f-e928-4892-9f2a-114e5d6fcd9b/d/I_6991/421 | Upload | 0.000000s | Ongoing | 0.000000s | 2020-Jun-10 15:19:20 | 8425472 |  |  | 2 | 0
83 | B_351/a9fc039f-e928-4892-9f2a-114e5d6fcd9b/d/I_7291/466 | Upload | 0.000000s | Ongoing | 0.000000s | 2020-Jun-10 15:19:20 | 8425472 |  |  | 3 | 0
83 | B_906/a9fc039f-e928-4892-9f2a-114e5d6fcd9b/d/I_9802/473 | Upload | 0.000000s | Ongoing | 0.000000s | 2020-Jun-10 15:19:20 | 8425472 |  |  | 4 | 0
```

3. Execute the `weka fs tier location` command to see a file's location status.
    a. `weka fs tier location <file path>`

Example:

```
weka fs tier location /mnt/simple/test0
Path             | File Type | File Size | Capacity in SSD (write-cache) | Capacity in SSD (read-cache) | Capacity in Object Storage
-----------------+-----------+-----------+-------------------------------+------------------------------+---------------------------
/mnt/simple/test0 | regular   | 7.97 GB   | 0 B                           | 0 B                          | 7.97 GB
```

Fetch a Tiered File Back to the Filesystem

1. Connect to a **Weka** console over SSH using a root/SUDO login.
2. Execute the `weka fs tier fetch` command.
    a. `weka fs tier fetch <path>`

Examples:

```
weka fs tier fetch /mnt/simple/test0
```

Create and Upload a Snapshot

3. Connect to a **Weka** console over SSH using a root/SUDO login.
4. Execute the `weka fs snapshot create` command.
    a. `weka fs snapshot create <file-system> <name> [<access-point>] [--source-snap=<source>] [--is-writable]`
5. Execute the `weka fs snapshot upload` command.
    a. `Weka fs snapshot upload <file-system> <snapshot>`

Examples:

```
weka fs snapshot create test_fs snapshot1
weka fs snapshot upload test_fs snapshot1
```

Restore a Snapshot to a New Filesystem

1. Connect to a **Weka** console over SSH using a root/SUDO login.
2. Execute the `weka fs download` command and restore the snapshot to a new filesystem.
    a. `weka fs download <name> <group-name> <total-capacity> <ssd-capacity> <obs> <locator>`

Examples:

```
weka fs download new_fs GROUP1 20T 5T cos 1
```

Appendix B: Troubleshooting

Verify Weka System Cluster Health

1. Connect to a **Weka** console over SSH using a root/SUDO login.
2. Execute the `weka cluster license` command.

```
[root@weka08 anchors]# weka status
WekaIO v3.7.0.5-beta (CLI build 3.7.0.5-beta)

       cluster:  (a9fc039f-e928-4892-9f2a-114e5d6fcd9b)
        status: OK (8 backends UP, 48 drives UP)
    protection: 5+2
     hot spare: 1 failure domains (13.46 TiB)
 drive storage: 94.28 TiB total, 42.69 TiB unprovisioned
         cloud: connected
       license: There's a problem with your license, please use `weka cluster license` to see
what is wrong

     io status: STARTED 16 days ago (96 io-nodes UP, 840 buckets UP)
    link layer: Ethernet
       clients: 0 connected
         reads: 0 B/s (0 IO/s)
        writes: 0 B/s (0 IO/s)
    operations: 0 ops/s
        alerts:
```

Verify Weka System Licensing

1. Connect to a **Weka** console over SSH using a root/SUDO login.
2. Execute the `weka cluster license` command.

```
[root@weka01 anchors]# weka cluster license
Licensing status: Classic

Current usage: 184356 GB raw drive capacity
               0 GB object-store capacity

Installed license: Valid from 2020-04-22T10:42:08Z
                   Expires at 2021-04-07T20:24:38Z
                   3000000 GB raw drive capacity
                   6000000 GB object-store capacity
```

Troubleshoot Object Store Configuration Errors

1. Confirm bucket name spelling including case.
2. Verify certificate subject name matches the hostname used.
3. Verify hostname can be resolved.

Appendix C: Test Data

Test data was random generated writes by Genset and Goader. Genset uses ASCII seed, and Goader is binary.

References

Weka System Documentation

- https://docs.weka.io/

Notices

This information was developed for products and services offered in the US. This material might be available from IBM® in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR
FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing IBM
Corporation
North Castle Drive, MD-NC119 Armonk,
NY 10504-1785
US

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The performance data discussed herein is presented as derived under specific operating conditions. Actual results may vary.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at Copyright and trademark information at www.ibm.com/legal/copytrade.shtml.

Accesser®, Cleversafe®, ClevOS™, Dispersed Storage®, dsNet®, IBM Cloud Object Storage Accesser®, IBM Cloud Object Storage Dedicated™, IBM Cloud Object Storage Insight™, IBM Cloud Object Storage Manager™, IBM Cloud Object Storage Slicestor®, IBM Cloud Object Storage Standard™, IBM Cloud Object Storage System™, IBM Cloud Object Storage Vault™, SecureSlice™, and Slicestor® are trademarks or registered trademarks of Cleversafe, an IBM Company and/or International Business Machines Corp.

Other product and service names might be trademarks of IBM or other companies.

IBM®

Printed in USA