Hitachi Content Software for File

4.0.x

User Guide

Hitachi Content Software for File is a high performance storage solution for AI, ML, analytics, and other GPU-accelerated workloads. It provides the speed of a distributed file system (DFS) with the capacity and hybrid cloud capabilities of an object store. The unique integration of file and object storage offers a tightly coupled, single solution for an appliance-like experience designed for ultra-high performance and capacity applications.

MK-HCSF000-02
October 2023

Contents

Preface

This document provides information and instructions for using the Hitachi Content Software for File (HCSF) system.

Please read this document carefully to understand how to use this product, and maintain a copy for your reference.

Intended audience

This document is intended for system administrators, Hitachi Vantara representatives, and authorized service providers who install, configure, and operate the HCSF system.

Readers of this document should be familiar with the following:

- Storage system and performance concepts, including clustering and networking.
- Storage array and tiering concepts.
- Object stores, including S3, Hitachi Content Platform, and Hitachi Content Platform for cloud scale.
- Data lifecycle management concepts.

Product version

This document revision applies to HCSF software version 4.1.*x* and later.

Release notes

Read the release notes before installing and using this product. They may contain requirements or restrictions that are not fully described in this document or updates or corrections to this document. Release notes are available on the Hitachi Vantara Support Website: https://knowledge.hitachivantara.com/Documents.

Document conventions

This document uses the following typographic conventions:

Convention	Description
Bold	▪ Indicates text in a window, including window titles, menus, menu options, buttons, fields, and labels. Example: Click **OK**. ▪ Indicates emphasized words in list items.
Italic	▪ Indicates a document title or emphasized words in text. ▪ Indicates a variable, which is a placeholder for actual text provided by the user or for output by the system. Example: `pairdisplay -g group` (For exceptions to this convention for variables, see the entry for angle brackets.)
`Monospace`	Indicates text that is displayed on screen or entered by the user. Example: `pairdisplay -g oradb`
< > angle brackets	Indicates variables in the following scenarios: ▪ Variables are not clearly separated from the surrounding text or from other variables. Example: `Status-<report-name><file-version>.csv` ▪ Variables in headings.
[ ] square brackets	Indicates optional values. Example: [a | b] indicates that you can choose a, b, or nothing.
{ } braces	Indicates required or expected values. Example: { a | b } indicates that you must choose either a or b.
| vertical bar	Indicates that you have a choice between two or more options or arguments. Examples: [a | b] indicates that you can choose a, b, or nothing. { a | b } indicates that you must choose either a or b.

This document uses the following icons to draw attention to information:

Icon	Label	Description
	Note	Calls attention to additional information.

Icon	Label	Description
	Tip	Provides helpful information, guidelines, or suggestions for performing tasks more effectively.
	Important	Highlights information that is essential to the completion of a task.
	Caution	Warns the user of adverse conditions and/or consequences (for example, disruptive operations, data loss, or a system crash).
	CAUTION	Warns the user of a hazardous situation that, if not avoided, could result in major or minor injury.
	WARNING	Warns the user of a hazardous situation which, if not avoided, could result in death or serious injury.

Conventions for storage capacity values

Physical storage capacity values (for example, disk drive capacity) are calculated based on the following values:

Physical capacity unit	Value
1 kilobyte (KB)	1,000 (10^3) bytes
1 megabyte (MB)	1,000 KB or $1{,}000^2$ bytes
1 gigabyte (GB)	1,000 MB or $1{,}000^3$ bytes
1 terabyte (TB)	1,000 GB or $1{,}000^4$ bytes
1 petabyte (PB)	1,000 TB or $1{,}000^5$ bytes
1 exabyte (EB)	1,000 PB or $1{,}000^6$ bytes

Logical capacity values (for example, logical device capacity, cache memory capacity) are calculated based on the following values:

Logical capacity unit	Value
1 block	512 bytes
1 cylinder	Mainframe: 870 KB

Logical capacity unit	Value
	Open-systems: ▪ OPEN-V: 960 KB ▪ Others: 720 KB
1 KB	1,024 (2^{10}) bytes
1 MB	1,024 KB or $1{,}024^2$ bytes
1 GB	1,024 MB or $1{,}024^3$ bytes
1 TB	1,024 GB or $1{,}024^4$ bytes
1 PB	1,024 TB or $1{,}024^5$ bytes
1 EB	1,024 PB or $1{,}024^6$ bytes

Accessing product documentation

Product user documentation is available on the Hitachi Vantara Support Website: https://knowledge.hitachivantara.com/Documents. Check this site for the most current documentation, including important updates that may have been made after the release of the product.

Getting help

The Hitachi Vantara Support Website is the destination for technical support of products and solutions sold by Hitachi Vantara. To contact technical support, log on to the Hitachi Vantara Support Website for contact information: https://support.hitachivantara.com/en_us/contact-us.html.

Hitachi Vantara Community is a global online community for Hitachi Vantara customers, partners, independent software vendors, employees, and prospects. It is the destination to get answers, discover insights, and make connections. **Join the conversation today!** Go to community.hitachivantara.com, register, and complete your profile.

Comments

Please send comments to doc.comments@hitachivantara.com. Include the document title and number, including the revision level (for example, -07), and refer to specific sections and paragraphs whenever possible. All comments become the property of Hitachi Vantara LLC.

Thank you!

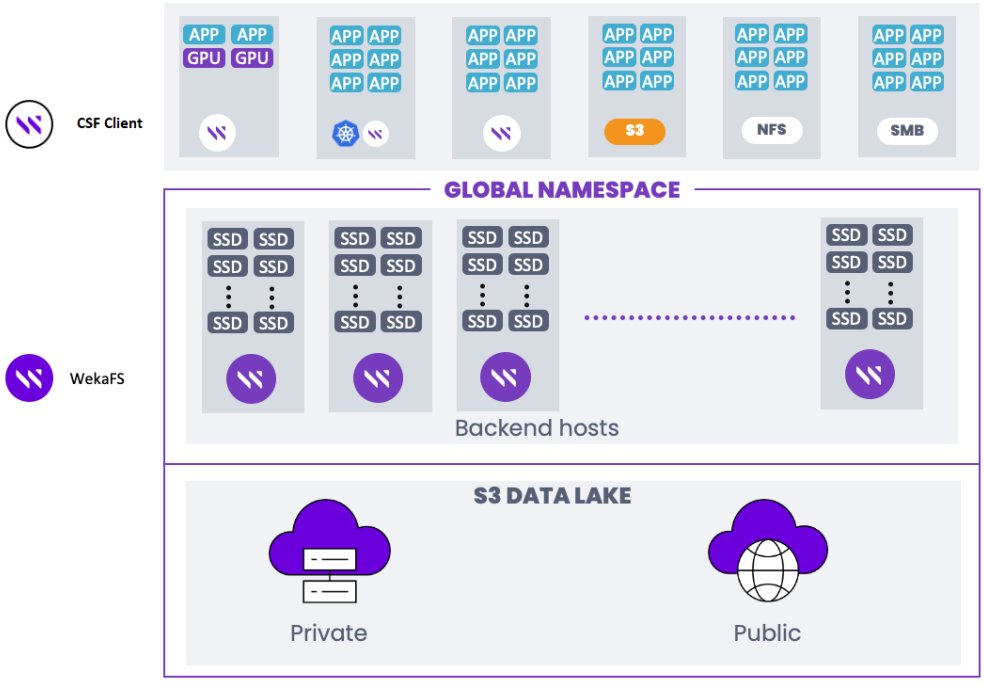
Chapter 1: About the Content Software for File system

The Content Software for File solution enables the implementation of a shareable, scalable, distributed file storage system.

Basic Content Software for File system deployment

The basic Content Software for File deployment model involves the creation of a shareable filesystem to be used by the application servers. This requires the installation of Content Software for File client software which implements a POSIX filesystem driver on each application server intended to access data. This filesystem driver enables each of the application servers to access the Content Software for File system as if it is a local drive, perceiving the Content Software for File system as a local attached filesystem device while it is actually shared among multiple application servers.

The file services are implemented by a group of backend hosts running the Content Software for File software and fully dedicated to the Content Software for File system. SSD drives for storing the data are installed on these servers. The resultant storage system is scalable to hundreds of backends and thousands of clients.



The Content Software for File backends are configured as a cluster which, together with the Content Software for File clients installed on the application servers, forms one large shareable, distributed and scalable file storage system:

Shareable

All clients can share the same filesystems, so any file written by any of the clients is immediately available to any client reading the data. In technical terms, this means that Content Software for File is a strongly-consistent, POSIX-compliant system.

Distributed

A Content Software for File system is formed as a cluster of multiple backends, each of which provides services concurrently.

Scalable

The Content Software for File system linear performance depends on the size of the cluster. Consequently, a certain amount of performance will be received for a cluster of size x, while doubling the size of the cluster to 2x will deliver double the performance. This applies to both data and metadata.

Features

The Content Software for File provides a number of unique features and functions.

Protection

The Content Software for File system is N+2 or N+4 fully protected, meaning that any 2 concurrent failures in drives or backends do not cause any loss of data and maintains the Content Software for File system up and running to provide continuous services. This is achieved through a complex distributed protection scheme, which is determined when forming a cluster. The data part can range from 3 to 16, and the protection scheme can be either 2 or 4, i.e., clusters can be 3+2, 10+2, and even 16+4 for a large cluster of backend hosts.

Distributed network scheme

The Content Software for File system implements an any-to-any protection scheme, ensuring that if a backend fails, a rebuild process is performed using all other backends, taking the data that resided on the failed backend and recreating it using redundancy on other backends in the cluster. Consequently, redundancy is not redundancy across groups of backends, but is achieved through groups of data sets that protect each other in the whole cluster of backends. In this way, if one backend fails in a cluster of 100 backends, all the other 99 backends will participate in the rebuild process, simultaneously reading and writing. This means that the Content Software for File system rebuild process is extremely fast, unlike traditional storage architectures where functioning backends are only a small part of the backends or drives participating in the rebuild process. Furthermore, the bigger the cluster, the faster the rebuild process.

Failed component replacement as a background process

The hot spare is configured in the Content Software for File system clusters by providing the extra capacity required to return to full redundancy after a rebuild, unlike traditional approaches which dedicate specific physical components as the hot spare. Consequently, a cluster of 100 backends will be configured with sufficient capacity to rebuild the data and return to full redundancy even following two failures, after which it is still possible to withstand another two failures.

This strategy for the replacement of a failed component does not affect the vulnerability of the system. Following a system failure, it is not necessary to replace a failed component with a valid component in order to recreate the data. In the Content Software for File system, data is immediately recreated, leaving the replacement of the failed component with a functioning component as a background process.

Failure domains

Failure domains are groups of backends that may fail because of a single, root cause. For example, all servers in a rack can be considered a failure domain if all are powered through a single power circuit, or all are connected through a single top-of-rack (TOR) switch. Consider a setup of 10 such racks with a cluster of 50 Content Software for File backends (five (5) backends in each rack). During formation of the cluster, it is possible to configure with 6+2 protection and make the Content Software for File system aware of these possible failure domains by forming a protection stripe across racks. In this way, the 6+2 stripe will be spread on different racks, ensuring that the system remains operational in a full rack failure and that data is not lost.

For failure domains, the stripe width must be less or equal to the failure domain count - if there are 10 racks and one of them represents a single point of failure, 16+4 cluster protection is not possible. Consequently, protection and support of failure domains is dependent on the stripe width, the protection level, and the number of hot spares required.

Prioritized data rebuild process

When a failure occurs, the data rebuild process begins by reading all the stripes where the failure occurred, rebuilding the data and returning to full protection. If a second failure occurs, there will actually be three possible types of stripes:

1. Stripes not affected by either of the failed components – no action required.
2. Stripes affected by only one of the failed components.
3. Stripes affected by both the failed components.

Naturally, according to rules of multiplicity, the number of stripes affected by two failed components is much smaller than the number of stripes affected by a single failed component. However, in situations where stripes affected by both the failed components have yet to be rebuilt, a third component failure will expose the Content Software for File system to data loss.

To reduce this risk, the Content Software for File system prioritizes the rebuild process, starting first with stripes affected by two component failures. Since the number of such stripes is much smaller, this rebuild process is performed very quickly, within minutes or less. The Content Software for File system then returns to the rebuild of stripes affected by only one failed component, and can still withstand another concurrent failure without any loss of data. This prioritized approach to the rebuild process ensures that data is almost never lost, and that service and data are always available.

Seamless distribution, scale, and performance

Each Content Software for File client installed on an application server directly accesses the relevant backend host storing the data, specifically that each client does not access one backend, which then forwards the access request. Content Software for File clients include a completely synchronized map of which backend stores which type of data, representing a joint configuration that all clients and backends are aware of.

When a Content Software for File client tries to access a certain file or an offset in a file, a cryptographic hash function indicates which backend contains the required file or offset. When a cluster expansion is performed or a component failure occurs, the backend responsibilities and capabilities are instantly redistributed between the various components. This is the basic mechanism that allows the Content Software for File system to linearly grow performance and is the key to linearly synchronizing scaling size to scaling performance. If, for example, backends are added to double the size of a cluster, different parts of the filesystems are redistributed to the new backends, thereby instantly delivering twice the performance.

Furthermore, if a cluster is just grown modestly for example, from 100 to 110 backends, it is not necessary to redistribute all the data, and only 10% of the existing data will be copied to the new backends, in order to equally redistribute the data on all the backends. This balancing of the data – extending participation of all backends in all read operations - is important for scaled performance, ensuring that there are no idle or full backends, and that each backend in a cluster stores the same amount of data.

The duration of all these completely seamless operations depends on the capacity of the root backends and the network bandwidth. Ongoing operations are not affected, and performance is improved as the redistribution of data is executed. Completion of the redistribution process delivers optimal capacity and performance.

Data reduction

Our enhanced data reduction maintains exceptional performance while delivering significant reductions on various workloads. The Content Software for File system looks for blocks of data that are similar to each other (they don’t need to be 100% identical like traditional data reduction techniques) and reduce them, storing any differences separately.

Data reduction can be enabled per filesystem. Compression ratios will be workload-dependent and are excellent with text-based data, large-scale unstructured datasets, log analysis, databases, code repositories, and sensor data. We are providing a Data Reduction Estimation Tool (DRET) that can run on existing file systems to calculate the reduction rate of your datasets. For more information, contact the Customer Success Team.

Converged Content Software for File system deployment

The Content Software for File system can be deployed in a converged configuration. An alternative to the basic system deployment, this enables the configuration of hundreds of application servers running user applications and installed with Content Software for File clients in order to access the cluster. Consequently, instead of provisioning servers fully dedicated to backends, it enables the installation of a client on each application server, and the installation of one or more SSDs as well as backend processes on the existing application servers. In such a configuration, the Content Software for File system backend processes operate as one big cluster, takeover the local SSDs and form a shareable, distributed and scalable filesystem available to the application servers, in the same way as in the basic system deployment. The only difference is that instead of installing SSDs on backends dedicated to the Content Software for File system, in this configuration the backends share the same physical infrastructure with the application servers.

This mixture of different storage and computation abilities delivers more effective performance and a better utilization of resources. However, unlike the basic Content Software for File system deployment, where an application server failure has no effect on the other backends, here the cluster will be affected if an application server is rebooted or fails. The cluster is still protected by the N+2 scheme, and can withstand two such concurrent failures. Consequently, converged Content Software for File deployments require more careful integration, as well as more detailed awareness between computation and storage management practices.

Otherwise, this is technically the same solution as the basic Content Software for File system deployment, with all the same system functionality features for protection, redundancy, failed component replacement, failure domains, prioritized data rebuilds and seamless distribution, scale and performance. Some of the servers may be installed with a Content Software for File backend process and a local SSD, while others may have clients only. This means that there can be a cluster of application servers with Content Software for File software installed on some and clients installed on others.

Selecting a redundancy scheme

Redundancy schemes in the Content Software for File system deployments can range from 3+2 to 16+4. There are a number of considerations for selecting the most suitable, optimal configuration. It all depends on redundancy, the data stripe width, the hot spare capacity, and the performance required during a rebuild from a failure.

Redundancy

Redundancy can be N+2 or N+4 and impacts both capacity and performance. A redundancy of 2 is sufficient for the majority of configurations. A redundancy of 4 is usually used for clusters of 100 or more backends, or for extremely critical data.

Data Stripe Width

The number of data components, which can be 3-16. The bigger the data stripe, the better the eventual net capacity. Consideration has to be given to both raw and net capacity. Raw capacity is the total capacity of SSDs in the deployment. Net capacity relates to how much is actually available for the storage of data. Consequently, bigger

stripe widths provide more net capacity but may impact performance under rebuild, as discussed below, in Performance Required During a Rebuild from a Failure. For extremely critical data, it is recommended to consult the Weka Support Team to determine whether the stripe width matches the resiliency requirements.

For extremely critical data, it is recommended to contact your customer support representative to determine whether the stripe width matches the resiliency requirements.

Note: The active failure domains count cannot be less than the stripe width, for example, a situation where two failure domains become unavailable in 3+2 protection with 6 failure domains, since this will leave the Content Software for File cluster vulnerable and unable to rebuild. In such situations, contact your customer support representative.

Hot Spare Capacity

An IT issue, relating to the time required to replace faulty components. The faster that IT succeeds in processing failures, or guarantees the replacement of faulty components, the lower the hot spare capacity required. The more relaxed, and hence cost-effective, the component replacement schedule is, the more the required hot spare capacity. For example, remotely-located systems visited once a quarter to replace any failed drives require more hot spares than systems with guaranteed 24/7 service.

Performance required during a rebuild from a failure

Is impacted only by read rebuild operations. Unlike other storage systems, writing performance is unaffected by failures and rebuilds, since Content Software for File systems continue writing to functioning backends in the cluster. However, read performance can be affected, because the reading of data from a failed component has to be performed from the whole stripe. This requires a simultaneous operation and an instant priority rebuild for data read operations. If, for example, one failure occurs in a cluster of 100 backends, performance will be affected by 1%; however, in a cluster of 100 backends with a stripe width of 16, performance will be reduced by up to 16% at the beginning of the rebuild. Naturally, the cluster size can exceed the stripe width or the number of failure domains. Consequently, for large clusters, it is recommended that the stripe width does not exceed 25% of the cluster size, e.g., for a cluster of 40 backends, 8+2 protection is recommended so that if a failure occurs, the impact on performance will not exceed 25%.

Write Performance

Is generally better the larger the stripe width since the system has to compute a smaller proportion of protected data to real data. This is particularly applicable to large writes in a system accumulating data for the first time

SSD capacity management

Terminologies relating to Content Software for File system capacity management and the formula for calculating the Content Software for File system net data storage capacity

Raw capacity

Raw capacity is the total capacity on all the SSDs assigned to a Content Software for File system cluster, e.g., 10 SSDs of 1 terabyte each have a total raw capacity of 10 terabytes. This is the total capacity available for the Content Software for File system. This will change automatically if more hosts or SSDs are added to the system.

Net capacity

Net capacity is the amount of space available for user data on the SSDs in a configured Content Software for File system. It is based on the raw capacity minus the Content Software for File filesystem overheads for redundancy protection and other needs. This will change automatically if more hosts or SSDs are added to the system.

Stripe width

The stripe width is the number of blocks that share a common protection set, which can range from 3 to 16. The Content Software for File system has distributed any-to-any protection. Consequently, in a system with a stripe width of 8, many groups of 8 data units spread on various hosts protect each other (rather than a group of 8 hosts forming a protection group). The stripe width is set during the cluster formation and cannot be changed. Stripe width choice impacts performance and space.

Note: If not configured, the stripe width is set automatically to #Failure Domains - Protection Level

Protection level

The protection level is the number of additional protection blocks added to each stripe, which can be either 2 or 4. A system with a protection level of 2 can survive 2 concurrent failures, while system data with a protection level of 4 is protected against any concurrent 4 host/disk failures, and its availability is protected against any 4 concurrent disk failures or 2 concurrent host failures. A large protection level has space and performance implications. The protection level is set during the cluster formation and cannot be changed.

Note: If not configured, the data protection drives in the cluster stripes are automatically set to 2.

Failure domain (optional)

A failure domain is a group of Content Software for File hosts, all of which can fail concurrently due to a single root cause, such as a power circuit or network switch failure. A cluster can be configured with explicit or implicit failure domains. For a system with explicit failure domains, each group of blocks that protect each other is spread on different failure domains. For a system with implicit failure domains, the group of blocks is spread on different hosts and each host is a failure domain. Additional failure domains can be added, and new hosts can be added to any existing or new failure domain.

Note: This documentation relates to a homogeneous Content Software for File system deployment, i.e., the same number of hosts per failure domain (if any), and the same SSD capacity per host. For information about heterogeneous Content Software for File system configurations, contact the customer support.

Hot spare

A hot spare is the number of failure domains that the system can lose, undergo a complete rebuild of data, and still maintain the same net capacity. All failure domains are always participating in storing the data, and the hot spare capacity is evenly spread within all failure domains.

The higher the hot spare count, the more hardware required to obtain the same net capacity. On the other hand, the higher the hot spare count, the more relaxed the IT maintenance schedule for replacements. The hot spare is defined during cluster formation and can be reconfigured at any time.

Note: If not configured, the hot spare is automatically set to 1.

Content Software for File filesystem overhead

After deducting the capacity for the protection and hot spares, only 90% of the remaining capacity can be used as net user capacity, with the other 10% of capacity reserved for the Content Software for File filesystems. This is a fixed formula that cannot be configured.

Provisioned capacity

The provisioned capacity is the total capacity assigned to filesystems. This includes both SSD and object store capacity.

Available capacity

The available capacity is the total capacity that can be used for the allocation of new filesystems, which is net capacity minus provisioned capacity.

Deductions from raw capacity to obtain net storage capacity

The net capacity of the Content Software for File system is obtained after the following three deductions performed during configuration:

1. Level of protection required, which is the amount of storage capacity to be dedicated for system protection.
2. Hot spare(s), that is the amount of storage capacity to be set aside for redundancy and to allow for rebuilding following a component failure.
3. Content Software for File filesystem overhead, in order to improve overall performance.

Formula for calculating SSD net storage capacity

$$\text{SSD Net Capacity} = \text{Raw Capacity} + \frac{\text{\# Failure Domains} - \text{Hot Spares}}{\text{\# Failure Domains}} + \frac{\text{Stripe Width}}{\text{Stripe Width} + \text{Protection}} + \text{File System Overhead}$$

Scenario 1

A homogeneous system of 10 hosts, each with 1 terabyte of Raw SSD Capacity, 1 hot spare, and a protection scheme of 6+2.

```
SSDNetCapacity = 10TB * (10-1) / 10 * 6/(6+2) * 0.9 = 6.075TB
```

Scenario 2

A homogeneous system of 20 hosts, each with 1 terabyte of Raw SSD Capacity, 2 hot spares, and a protection scheme of 16+2.

```
SSDNetCapacity = 20TB * (20-2) / 20 * 16/(16+2) * 0.9 = 14.4TB
```

Filesystems, object stores, and filesystem groups

There are three types of entities relevant to data storage in the Content Software for File system: filesystems, object stores, and filesystem groups.

About filesystems

A Content Software for File filesystem is similar to a regular on-disk filesystem while distributed across all the hosts in the cluster. Consequently, filesystems are not associated with any physical object in the Content Software for File system and act as root directories with space limitations.

The system supports a total of up to 1024 filesystems. All of which are equally balanced on all SSDs and CPU cores assigned to the system. This means that the allocation of a new filesystem or resizing a filesystem are instant management operations performed without any constraints.

A filesystem has a defined capacity limit and is associated with a predefined filesystem group. A filesystem that belongs to a tiered filesystem group must have a total capacity limit and an SSD capacity cap. All filesystems' available SSD capacity cannot exceed the total SSD net capacity.

Thin provisioning

Thin provisioning is a method of on-demand SSD capacity allocation based on user requirements. In thin provisioning, the filesystem capacity is defined by a minimum guaranteed capacity and a maximum capacity (virtually can be more than the vailable SSD capacity).

The system allocates more capacity (up to the total available SSD capacity) for users who consume their allocated minimum capacity. Alternatively, when they free up space by deleting files or transferring data, the idle space is reclaimed, repurposed, and used for other workloads that need the SSD capacity.

Thin provisioning is beneficial in various use cases:

- Tiered filesystems: On tiered filesystems, available SSD capacity is leveraged for extra performance and released to the object store once needed by other filesystems.
- Auto-scaling groups: When using auto-scaling groups, thin provisioning can help to automatically expand and shrink the filesystem's SSD capacity for extra performance.
- Separation of projects to filesystems: If it is required to create a separate filesystem for each project, and the administrator doesn't expect all filesystems to be fully utilized simultaneously, creating a thin provisioned filesystem for each project is a good solution. Each filesystem is allocated with a minimum capacity but can consume more when needed based on the actual available SSD capacity.

Filesystem limits

- Number of files or directories: Up to 6.4 trillion (6.4 * 10^12)
- Number of files in a single directory: Up to 6.4 billion (6.4 * 10^9)
- Total capacity with object store: Up to 14 EB
- Total SSD capacity: Up to 512 PB
- File size: UP to 4 PB

Encrypted filesystems

Both data at rest (residing on SSD and object store) and data in transit can be encrypted. This is achieved by enabling the filesystem encryption feature. A decision on whether a filesystem is to be encrypted is made when creating the filesystem.

To create encrypted filesystems, deploy a Key Management System (KMS).

Note: You can only set the data encryption when creating a filesystem.

Metadata limitations

In addition to the capacity limitation, each filesystem has a limitation on the amount of metadata. The system-wide metadata limit is determined by the SSD capacity allocated to the Content Software for File system and the RAM resources allocated to the Content Software for File system processes.

The Content Software for File system keeps tracking metadata units in the RAM. If it reaches the RAM limit, it pages these metadata tracking units to the SSD and alerts. This leaves enough time for the administrator to increase system resources, as the system keeps serving IOs with a minimal performance impact.

By default, the metadata limit associated with a filesystem is proportional to the filesystem SSD size. It is possible to override this default by defining a filesystem-specific max-files parameter. The filesystem limit is a logical limit to control the specific filesystem usage and can be updated by the administrator when necessary.

The total metadata limits for all the filesystems can exceed the entire system metadata information that can fit in the RAM. For minimal impact, in such a case, the least-recently-used units are paged to disk, as necessary.

Metadata units calculation

Each metadata unit consumes 4 KB of SSD space (not tiered) and 20 bytes of RAM.

Throughout this documentation, the metadata limitation per filesystem is referred to as the max-filesparameter, which specifies the number of metadata units (not the number of files). This parameter encapsulates both the file count and the file sizes.

The following table specifies the required number of metadata units according to the file size. These specifications apply to files residing on SSDs or tiered to object stores.

File size	Number of metadata units	Example
< 0.5 MB	1	A filesystem with 1 billion files of 64 KB each requires 1 billion metadata units.
0.5 MB - 1 MB	2	A filesystem with 1 million files of 750 KB each, requires 2 million metadata units.
> 1 MB	2 for the first 1 MB plus 1 per MB for the rest MBs	▪ A filesystem with 1 million files of 129 MB each requires 130 million metadata units. 2 units for the first 1 MB plus 1 unit per MB for 128 MB. ▪ A filesystem with 10 million files of 1.5 MB each requires 30 million units. ▪ A filesystem with 10 million files of 3 MB each requires 40 million units.

Note: Each directory requires two metadata units instead of one for a small file.

About object stores

In the Content Software for File system, object stores represent an optional external storage media, ideal for storing warm data. Object stores used in tiered Content Software for File system configurations can be cloud-based, located in the same location (local), or at a remote location.

Content Software for File supports object stores for tiering (tiering and local snapshots) and backup (snapshots only). Both tiering and backup can be used for the same filesystem.

Using object store buckets optimally is achieved when a cost-effective data storage tier is required at a price point that cannot be satisfied by server-based SSDs.

An object store bucket definition contains the object store DNS name, bucket identifier, and access credentials. The bucket must be dedicated to the Content Software for File system and not be accessible by other applications.

Filesystem connectivity to object store buckets can be used in the data lifecycle management and Snap-to-Object features.

About filesystem groups

In the Content Software for FileContent Software for File system, filesystems are grouped into a maximum of eight filesystem groups.

Each filesystem group has tiering control parameters. While tiered filesystems have their object store, the tiering policy is the same for each tiered filesystem under the same filesystem group.

For information on managing these entities, see Managing Filesystems, Object Stores, and Filesystem Groups (on page 52).

networking

This page reviews the theory of operation for Content Software for File networking.

Overview

The Content Software for File system supports the following types of networking technologies:

- InfiniBand (IB)
- Ethernet

The currently-available networking infrastructure dictates the choice between the two. If a Content Software for File cluster is connected to both infrastructures, it is possible to connect Content Software for File clients from both networks to the same cluster.

The Content Software for File system networking can be configured either as performance-optimized, where the CPU cores are dedicated to Content Software for File and the use of DPDK networking takes place and cores, or, as CPU-optimized where cores are not dedicated and we use either DPDK (when supported by the NIC drivers) or in-kernel networking (UDP mode).

Performance-optimized networking (DPDK)

For performance-optimized networking, the Content Software for File system does not use standard kernel-based TCP/IP services, but a proprietary infrastructure based on the following:

- Use of DPDK to map the network device in the user space and make use of the network device without any context switches and with zero-copy access. This bypassing of the kernel stack eliminates the consumption of kernel resources for networking operations and can be scaled to run on multiple hosts. It applies to both backend and client hosts and enables the Weka system to fully saturate 200 GB links.
- Implementation of a proprietary Content Software for File protocol over UDP, meaning that the underlying network may involve routing between subnets or any other networking infrastructure that supports UDP.

The use of DPDK delivers operations with extremely low-latency and high throughput. Low latency is achieved by bypassing the kernel and sending and receiving packages directly from the NIC. High throughput is achieved because multiple cores in the same host can work in parallel, without a common bottleneck.

Before proceeding, it is important to understand several key terms used in this section, namely DPDK, SR-IOV.

DPDK

Data Plane Development Kit (DPDK) is a set of libraries and network drivers for highly efficient, low latency packet processing. This is achieved through several techniques, such as kernel TCP/IP bypass, NUMA locality, multi-core processing, and device access via polling to eliminate the performance overhead of interrupt processing. In addition, DPDK ensures transmission reliability, handles retransmission, and controls congestion.

DPDK implementations are available from several sources. OS vendors such as Redhat and Ubuntu provide their DPDK implementations through their distribution channels. Mellanox OpernFabrics Enterprise Distribution for Linux (Mellanox OFED), which is a suite of libraries, tools, and drivers supporting Mellanox NICs, offers its own DPDK implementation.

SR-IOV

Single Root I/O Virtualization (SR-IOV) is an extension to the PCI Express (PCIe) specification that enables PCIe virtualization. It works by allowing a PCIe device, such as a network adapter, to appear as multiple PCIe devices, or functions. There are two categories of functions - Physical Function (PF) and Virtual Function (VF). PF is a full-fledged PCIe function that can also be used for configuration. VF is a virtualized instance of the same PCIe device and is created by sending appropriate commands to the device PF. Typically, there are many VFs, but only one PF per physical PCIe device. Once a new VF is created, it can be mapped by an object such as a virtual machine, container, or, in the Weka system, by a 'compute' process.

SR-IOV technology should be supported by both the software and hardware to take advantage of it. Software support is included in the Linux kernel, as well as the Content Software for File system software. Hardware support is provided by the computer BIOS and the network adapter but is usually disabled out of the factory. Consequently, it should be enabled before installing the Content Software for File system software.

CPU-optimized networking

For CPU-optimized networking Content Software for File can yield CPU resources to other applications. That is useful when the extra CPU cores are needed for other purposes. However, the lack of CPU resources dedicated to the Weka system comes with the expense of reduced overall performance.

DPDK without core dedication

For CPU-optimized networking, when mounting filesystems using statless clients, it is possible to use DPDK networking without dedicating cores. This mode is recommended when available and supported by the NIC drivers. In this mode, the DPDK networking uses RX interrupts instead of dedicating the cores.

Note: This mode is supported in most NIC drivers, but not in all, consult https://doc.dpdk.org/guides-18.11/nics/overview.html for compatibility.

AWS (ENA drivers) does not support this mode, hence for CPU-optimized networking in AWS use the UDP Mode.

UDP mode

Content Software for File can also use in-kernel processing and UDP as the transport protocol. This mode of operation is commonly referred to as the 'UDP mode'.

Since the UPD-mode uses in-kernel processing, it is compatible with older platforms lacking the support of kernel offloading technologies (DPDK) or virtualization (SR-IOV), as legacy hardware such as the Mellanox CX3 family of NICs.

Data lifecycle management

The principles of data lifecycle management and how data storage is managed in SSD-only and tiered Content Software for File system configurations.

Media options for data storage in the Content Software for File system

In the Content Software for File system, data can be stored on two forms of media:

1. On locally-attached SSDs, which are an integral part of the Content Software for File system configuration.
2. On object-store systems external to the Content Software for File system, which are either third-party solutions, cloud services, or part of the Content Software for File system.

The Content Software for File system can be configured either as an SSD-only system or as a data management system consisting of both SSDs and object stores. By nature, SSDs provide high performance and low latency storage, while object stores compromise performance and latency but are the most cost-effective solution available for storage. Consequently, users focused on high performance only should consider using an SSD-only Content Software for File system configuration, while users are seeking to balance performance and cost should consider a tiered data management system, with the assurance that the Content Software for File system features will control the allocation of hot data on SSDs and warm data on object stores, thereby optimizing the overall user experience and budget.

Note: In SSD-only configurations, the Content Software for File system will sometimes use an external object store for backup, as explained in Snap-To-Object Data Lifecycle Management (on page 90).

Guidelines for data storage in tiered Content Software for File system configurations

In tiered Content Software for File system configurations, there are various locations for data storage as follows:

1. Metadata is stored only on SSDs.
2. Writing of new files, adding data to existing files, or modifying the content of files is always terminated on the SSD, irrespective of whether the file is currently stored on the SSD or tiered to an object-store.
3. 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).

This data management approach to data storage on one of two possible media requires system planning to ensure that most commonly-used data (hot data) resides on the SSD to ensure high performance, while less-used data (warm data) is stored on the object store. In the Content Software for File system, this determination of the data storage media is a completely seamless, automatic, and transparent process, with users and applications unaware of the transfer of data from SSDs to object stores, or from object stores to SSDs. The data is accessible at all times through the same strongly-consistent POSIX filesystem API, irrespective of where it is stored. Only latency, throughput, and IOPS are affected by the actual storage media.The network resources allocated to the object store connections can be controlled. This enables cost control when using cloud-based object storage services since the cost of data stored in the cloud depends on the quantity stored and the number of requests for access made.

Furthermore, the Weka system tiers data in chunks, rather than complete files. This enables the smart tiering of subsets of a file (and not only complete files) between SSDs and object stores.

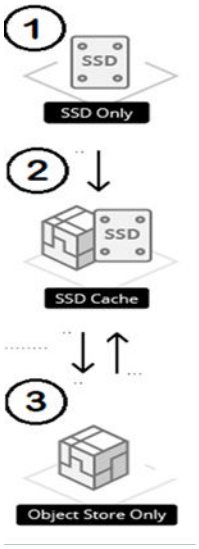
The network resources allocated to the object store connections can be controlled. This enables cost control when using cloud-based object storage services since the cost of data stored in the cloud depends on the quantity stored and the number of requests for access made.

States in the Content Software for File system data management storage process

Data management represents the media being used for the storage of data. In tiered Content Software for File system configurations, data can exist in one of three possible states:

1. SSD-only: When data is created, it exists only on the SSDs.
2. SSD-cached: A tiered copy of the data exists on both the SSD and the object store.
3. Object Store only: Data resides only on the object store.

Note: These states represent the lifecycle of data, and not the lifecycle of a file. When a file is modified, each modification creates a separate data lifecycle for the modified data.



The Data Lifecycle Diagram represents the transitions of data between the above states. #1 represents the Tiering operation, #2 represents the Releasing operation, and #3 represents the Rehydrating operation:

1. Tiering of data from the SSD to create a replicate in the object store. A guideline for the tiering of data is based on a user-defined, time-based policy *Tiering Cue*.
2. Releasing data from the SSD, leaving only the object store copy (based on the demand for more space for data on the SSD). A guideline for the release of data is based on a user-defined, time-based policy *Retention Period*.
3. Rehydrating data from the object store to the SSD for the purpose of data access.

In order to read data residing only on an object store, the data must first be rehydrated back to the SSD.

In the Content Software for File system, file modification is never implemented as in-place write, but rather as a write to a new area located on the SSD, and the relevant modification of the meta-data. Consequently, write operations are never associated with object store operations.

The role of SSDs in tiered Content Software for File configurations

All writing in the Content Software for File system is performed to SSDs. The data residing on SSDs is hot data, that is data that is currently in use. In tiered Content Software for File configurations, SSDs have three primary roles in accelerating performance: metadata processing, a staging area for writing, and as a cache for read performance.

Metadata processing

Since filesystem metadata is by nature a large number of update operations each with a small number of bytes, the embedding of metadata on SSDs serves to accelerate file operations in the Content Software for File system.

SSD as a staging area

Since writing directly to an object store demands high latency levels while waiting for approval that the data has been written, with the Content Software for File system there is no writing directly to object stores. Much faster writing is performed directly to the SSDs, with very low latency and therefore much better performance. Consequently, in the system, the SSDs serve as a staging area, providing a buffer that is big enough for writing until later tiering of data to the object store. On completion of writing, the Content Software for File system is responsible for tiering the data to the object store and for releasing it from the SSD.

SSD as a cache

Recently accessed or modified data is stored on SSDs, and most read operations will be of such data and served from SSDs. This is based on a single, large least recently used (LRU) clearing policy for the cache that ensures optimal read performance.

Note: On a tiered filesystem, the total capacity determines the maximum capacity that will be used to store data. It could be that it will all reside on the object store due to the SSD uses above and the below time-based policies.

For example, consider a 100 TB filesystem (total capacity) with a 10TB SSD capacity for this filesystem. It could be that all the data will reside on the object-store, and no new writes will be allowed, although the SSD space is not completely used (until deleting files or increasing filesystem total size), leaving the SSD for metadata and cache only.

Time-based policies for the control of a data storage location

The Content Software for File system includes user-defined policies which serve as guidelines to control the data storage management. They are derived from a number of factors:

1. The rate at which data is written to the system and the quantity of data.
2. The capacity of the SSDs configured to the Content Software for File system.
3. The speed of the network between the Content Software for File system and the object store, and the performance capabilities of the object store itself, for example, how much the object store can actually contain.

Filesystem groups are used to define these policies, while a filesystem is placed in a filesystem group according to the desired policy if the filesystem is tiered.

For tiered filesystems, the following parameters should be defined per filesystem:

1. The size of the filesystem.
2. The amount of filesystem data to be stored on the SSD.

The following parameters should be defined per filesystem group:

1. The *Data Retention Period Policy*, a time-based policy which is the target time for data to be stored on an SSD after creation, modification or access, and before release from the SSD, even if it is already tiered to the object store, for metadata processing and SSD caching purposes (this is only a target; the actual release schedule depends on the amount of available space).
2. The *Tiering Cue Policy*, a time-based policy which determines the minimum amount of time that data will remain on an SSD before it is considered for release to the object store. As a rule of thumb, this should be configured to a third of the Retention Period, and in most cases, this will work well. The Tiering Cue is important because it is pointless to tier a file which is about to be modified or deleted from the object store.

For example, when writing log files which are processed every month but retained forever, it is recommended to define a Retention Period of 1 month, a Tiering Cue of 1 day, and ensure that there is sufficient SSD capacity to hold 1 month of log files.

When storing genomic data which is frequently accessed during the first 3 months after creation, requires a scratch space for 6 hours of processing, and requires output to be retained forever: It is recommended to define a Retention Period of 3 months and to allocate an SSD capacity that will be sufficient for 3 months of output data and the scratch space. The Tiering Cue should be defined as 1 day, in order to avoid a situation where the scratch space data is tiered to an object store and released from the SSD immediately afterwards.

Note: Using the *Snap-To-Object* feature causes data to be tiered regardless of the tiering policies. Snap-To-Object enables all the data of a specific snapshot (including metadata and every file) to be committed to an object store.

Bypassing the time-based policies

Regardless of the time-based policies, it is possible to use a special mount option `obs_direct` to bypass the time-based policies. Any creation or writing of files from a mount point with this option will mark it to release as soon as possible, before taking into account other files retention policy.

For more information, see Advanced Data Lifecycle Management (on page 68)

Content Software for File client and mount modes

Understanding the Content Software for File system client and possible mount modes of the operation in relation to the page cache.

The Content Software for File system client

The Content Software for File system client is a standard, POSIX-compliant filesystem driver installed on application servers that enable file access to the filesystems. Similar to any other filesystem driver, the system client intercepts and executes all filesystem operations. This enables the Content Software for File system to provide applications with local filesystem semantics and performance (as opposed to NFS mounts) while providing a centrally managed, shareable resilient storage.

The Content Software for File system client is tightly integrated with the Linux operating system page cache, which is a transparent caching mechanism that stores parts of the filesystem content in the client host RAM. The operating system maintains a page cache in unused RAM capacity of the application server, delivering quick access to the contents of the cached pages and overall performance improvements.

The page cache is implemented in the Linux kernel and is fully transparent to applications. All physical memory not directly allocated to applications is used by the operating system for the page cache. Since the memory would otherwise be idle and is easily reclaimed when requested by applications, there is usually no associated performance penalty and the operating system might even report such memory as "free" or "available". For a more detailed description of the page cache, see Page Cache, the Affair Between Memory and Files.

The Content Software for File client can control the information stored in the page cache and also invalidate it, if necessary. Consequently, the system can utilize the page cache for cached high-performance data access while maintaining data consistency across multiple hosts.

Each filesystem can be mounted in one of two modes of operation in relation to the page cache:

Read cache
Used where only read operations are using the page cache, file data is coherent across hosts, and resilient to client failures.

Write cache (default)
Used where both read and write operations are using the page cache while keeping data coherency across hosts, which provides the highest data performance.

Note: Symbolic links are always cached in all cached modes.

Note: Unlike actual file data, the file metadata is managed in the Linux operation system by the *Dentry (directory entry) cache*, which maximizes efficiency in the handling of directory entries, and is not strongly consistent across Content Software for File client hosts. At the cost of some performance compromise, metadata can be configured to be strongly consistent by mounting without Dentry cache (using `dentry_max_age_positive=0`, `dentry_max_age_negative=0` mount options) if metadata consistency is critical for the application.

Read cache mount mode

When mounting in this mode, the page cache uses write cache in the write-through mode; so, any write is acknowledged to the customer application only after being safely stored on resilient storage. This applies to both data and metadata writes. Consequently, only read operations are accelerated by the page cache.

In the Content Software for File system, by default, any data read or written by customer applications is stored on a local host read page cache. As a shareable filesystem, the Content Software for File system monitors whether another host tries to read or write the same data and if necessary, invalidates the cache entry. Such invalidation may occur in two cases:

- If a file that is being written by one client host is currently being read or written by another client host.
- If a file that is being read by one host is currently being written from another host.

This mechanism ensures coherence, providing the Content Software for File system with full page cache utilization whenever only a single host or multiple hosts access a file for read-only purposes. If multiple hosts access a file and at least one of them is writing to the file, the page cache is not used and any IO operation is handled by the backends. Conversely, when either a single host or multiple hosts open a file for read-only purposes, the page cache is fully utilized by the Content Software for File client, enabling read operations from memory without accessing the backend hosts.

Note: A host is defined as writing to a file on the actual first write operation, and not based on the read/write flags of the open system call.

Note: In some scenarios, particularly random reads of small blocks of data from large files, a read cache enablement can create an amplification of reads, due to the Linux operating system prefetch mechanism. For details about this scenario, see Understanding the <bdi> identifier.

Write cache mount mode (default)

In this mount mode, the Linux operating system is used as write-back, rather than write-through; specifically, the write operation is acknowledged immediately by the Content Software for File client and is stored in resilient storage as a background operation.

This mode can provide significantly more performance, particularly in relation to write latency, while keeping data coherency; meaning, if a file is accessed through another host, it invalidates the local cache, and syncs the data to get a coherent view of the file.

To sync the filesystem and commit all changes in the write cache, use the following system calls: `sync`, `syncfs`, and `fsync`.

Multiple mounts on a single host

The Content Software for File client supports multiple mount points of the same file system on the same host, even with different mount modes. This can be effective in environments such as containers where different processes in the host need to have different definitions of read/write access or caching schemes.

Note: Two mounts on the same hosts are treated as two different hosts with respect to the consistency of the cache, as described above. So, for example, two mounts on the same host, mounted with write cache mode might have different data at the same point in time.

Key terms

Term	Description
Agent	The Content Software for File agent is software installed on user application servers that need access to the Content Software for File file services. When using the Stateless Client feature, the agent is responsible for ensuring that the correct client software version is installed (depending on the cluster version) and that the client connects to the correct cluster.
Backend Host	A host that runs the Content Software for File software and is installed with SSD drives dedicated to the Content Software for File system, providing services to client hosts. A group of backend hosts forms a storage cluster.
Client	The Content Software for File client is software installed on user application servers that need access to Content Software for File file services. The Content Software for File client implements a kernel-based filesystem driver and the logic and networking stack to connect to the Content Software for File backend hosts and be part of a cluster. In general industry terms, "client" may also refer to an NFS, SMB, or S3 client that uses those protocols to access the Content Software for File filesystem. For NFS, SMB, and S3 the Content Software for File client is not required to be installed in conjunction with those protocols.
Cluster	A collection of Content Software for File backend hosts, together with Content Software for File clients installed on the application servers, forming one sharable, distributed, and scalable file storage system.
Container	Content Software for File uses Linux containers (LXC) as the mechanism for holding one node or keeping multiple nodes together. Containers can have different nodes within them. They can have frontend nodes and associated DPDK libraries within the container, or backend nodes, drive nodes, management node, and DPDK libraries, or can have NFS, SMB, or S3 services nodes running within them. A host can have multiple containers running on it at any time.

Term	Description
Converged Deployment	A Content Software for File configuration in which Content Software for File backend nodes run on the same host with applications.
Data Retention Period	The target period of time for tiered data to be retained on an SSD.
Data Stripe Width	The number of data blocks in each logical data protection group.
Dedicated Deployment	A Content Software for File configuration that dedicates complete servers and all of their allocated resources to Content Software for File backends, as opposed to a converged deployment.
Failure Domain	A collection of hardware components that can fail together due to a single root cause.
Filesystem Group	A collection of filesystems that share a common tiering policy to object-store.
Frontend	Is the collection of Content Software for File software that runs on a client and accesses storage services and IO from the Content Software for File storage cluster. The frontend consists of a frontend node that delivers IO to the Content Software for File driver, a DPDK library, and the Content Software for File POSIX driver.
Hot Data	Frequently used data (as opposed to warm data), usually residing on SSDs.
Host	A physical or virtual server that has hardware resources allocated to it and software running on it that provides compute or storage services. Content Software for File uses backend hosts in conjunction with clients to deliver storage services. In general industry terms, in a cluster of hosts, sometimes "nodes"is used instead.
Net Capacity	Amount of space available for user data on SSDs in a configured Content Software for File system.

Term	Description
Node	A software instance that Content Software for File uses to run and manage WekaFS. Nodes are dedicated to managing different functions such as (1) NVMe Drives and IO to the drives, (2) compute nodes for filesystems and cluster-level functions and IO from clients, (3) frontend nodes for POSIX client access and sending IO to the backend nodes, and (4) management nodes for managing the overall cluster. In general industry terms, a node also may be referenced as a discrete component in a hardware or software cluster. Sometimes when referring to hardware, the term host may be used instead.
POSIX	The Portable Operating System Interface (POSIX) is a family ofstandards specified by the
Provisioned Capacity	The total capacity that is assigned to filesystems. This includes both SSD and object store capacity.
Prefetch	The Content Software for File process of rehydrating data from an object store to an SSD, based on a prediction of future data access.
Raw Capacity	Total SSD capacity owned by the user.
Retention Period	The target time for data to be stored on SSDs before releasing from the SSDs to an object-store.
Releasing	The deletion of the SSD copy of data that has been tiered to the object-store.
Rehydrating	The creation of an SSD copy of data stored only on the object-store.
Server	In Content Software for File terms, a physical or virtual instantiation of hardware on which software runs and provides compute or storage services. In general industry terms, a server may also refer to a software process that provides a service to another process, whether on the same host or to a client (e.g. NFS server, SMB server, etc.).
Stem Mode	A mode of the Content Software for File software that has been installed and is running, but has not been attached to a cluster.
Snap-To-Object	A Content Software for File feature for uploading snapshots to object stores.
Tiered Content Software for File Configuration	Content Software for File configuration consisting of SSDs and object stores for data storage.

Term	Description
Tiering	Copying of data to an object store, while it still remains on the SSD.
Tiering Cue	The minimum time to wait before considering data for tiering from an SSD to an object-store.
Unprovisioned Capacity	The storage capacity that is available for new filesystems.
VF	Virtual Function
Warm Data	Less frequently-used data (as opposed to hot data), usually residing on an object-store.

Chapter 2: Typical Content Software for File configuration

Product configuration is described for:

- Backend hosts (on page 40).
- Backend hosts with DPDK-supporting Mellanox NICs (on page 40).

Backend hosts

In a typical Content Software for File system configuration, the backend hosts access the network function in two different methods:

- Standard TCP/UDP network for management and control operations.
- High-performance network for data-path traffic.

Note: To run both functions on the same physical interface, contact your customer support representative.

The high-performance network used to connect all the backend hosts must be DPDK-based. This internal Weka network also requires a separate IP address space (see Network Planning and Network Configuration). For this, the Weka system maintains a separate ARP database for its IP addresses and virtual functions and does not use the kernel or operating system ARP services.

Backend hosts with DPDK-supporting Mellanox and Intel E810 NICs

For backend hosts equipped with DPDK-supporting Mellanox (CX-4 or newer) and Intel E810 NICs, the following conditions must be met:

- Mellanox OFED must be installed and loaded.
- There is no need to use SR-IOV, so the number of IPs allocated to the backend hosts on the internal network should be the total number of backend hosts, i.e., 8 IPs for 8 backend hosts (using the example above).

Note: SR-IOV enablement in the hardware is optional. If enabled, DPDK generates its own MAC addresses for the VFs (Virtual Functions) of the NIC and the same NIC can support multiple MAC addresses, some handled by the operating system and others by the Weka system.

Backend hosts with DPDK-supporting the other NICs

For backend hosts equipped with DPDK-supporting the other NICs, the following conditions must be met:

- A driver with DPDK support must be installed and loaded.
- SR-IOV must be enabled in the hardware (BIOS + NIC).
- The number of IPs allocated to the backend hosts on the internal network should be the total number of Weka software processes plus the total number of backend hosts. For example, a cluster consisting of 8 machines running 10 Weka processes each requires 88 (80 + 8) IPs on the internal network. The IP requirements for the Weka clients are outlined below in the Client Hosts section.

Client hosts

Unlike Content Software for File backend nodes that must be DPDK/SR-IOV based, the Content Software for File client hosts (application servers) can use either DPDK-based or UDP modes. The DPDK mode is the natural choice for the newer, high-performing platforms that support it.

Client hosts with DPDK-supporting Mellanox and Intel E810 NICs

For client hosts equipped with DPDK-supporting Mellanox (CX-4 or newer) and Intel E810 NICs, the following conditions must be met:

- Mellanox OFED must be installed and loaded.
- There is no need to use SR-IOV, so the number of IPs allocated to the client hosts on the internal network should be the total number of client hosts, i.e., 10 IPs for 10 client hosts (using the example above).

Client hosts with DPDK-supporting the other NICs

For client hosts equipped with DPDK-supporting the other NICs, the following conditions must be met to use the DPDK mode:

- A driver with DPDK support must be installed and loaded.
- SR-IOV must be enabled in the hardware (BIOS + NIC).
- The number of IPs allocated to the Intel client hosts on the internal network should be the total number of Weka system FrontEnd (FE) processes (typically no more than 2 per host) plus the total number of client hosts. For example, 10 client hosts with 1 FE process per client require 20 IPs (10 FE IPs + 10 IPs).

Client hosts in UDP mode

The UDP mode is available for legacy clients lacking SR-IOV or DPDK support, or where there is no requirement for low latency, high throughput IO.

For client hosts in the UDP mode, the following conditions must be met:

- The native driver must be installed and loaded.
- The number of IPs allocated to the client hosts on the internal network should be equal to the total number of client hosts. For example, 10 client hosts in the UDP mode require 10 IPs on the internal network.

High availability (HA)

For HA support, the Content Software for File system must be configured with no single component representing a single point of failure. Multiple switches are required, and hosts must have one leg on each switch.

HA for hosts is achieved either through the implementation of two network interfaces on the same host or via LACP (ethernet only, modes 1 and 4). Using a non-LACP approach sets a redundancy that enables the Weka software to utilize two interfaces for HA and bandwidth, respectively.

HA performs failover and failback for reliability and load balancing on both interfaces and is operational for both Ethernet and InfiniBand. If not using LACP, it requires doubling the number of IPs on both the host and the IO nodes.

When working with HA networking, it is useful to hint the system (using the label parameter in weka cluster host net add command to identify the switch a network port is connected to) to send data between hosts through the same switch rather than using the ISL or other paths in the fabric. This can reduce the overall traffic in the network.

Note: LACP is currently supported between ports on a single Mellanox NIC, and is not supported when using VFs.

RDMA and GPUDirect storage

GPUDirect Storage enables a direct data path between storage and GPU memory. GPUDirect Storage avoids extra copies through a bounce buffer in the CPU's memory. It allows a direct memory access (DMA) engine near the NIC or storage to move data on a direct path into or out of GPU memory without burdening the CPU or GPU.

When enabled, the Content Software for File system automatically utilizes the RDMA data path and GPUDirect Storage in supported environments. When the system identifies it can use RDMA, both in UDP and DPDK modes, it utilizes the use for workload it can benefit from RDMA (with regards to IO size: 32K+ for reads and 256K+ for writes).

Using RDMA/GPUDirect Storage, it is thus possible to get a performance gain. You can get much higher performance from a UDP client (which does not require dedicating a core to the Content Software for File system), get an extra boost for a DPDK client, or assign fewer cores for the Content Software for File system in the DPDK mode to get the same performance.

Limitations

For the RDMA/GPUDirect Storage technology to take into effect, the following requirements must be met:

- All the cluster hosts must support RDMA networking
- For a client host:
  - GPUDirect Storage - the IB interfaces added to the Nvidia GPUDirect configuration should support RDMA
  - RDMA - all the NICs used by Weka must support RDMA networking
- Encrypted filesystems: The framework will not be utilized for encrypted filesystems and will fall back to work without RDMA/GPUDirect for IOs to encrypted filesystems
- A NIC is considered to support RDMA Networking if the following requirements are met:
  - For GPUDirect Storage only: InfiniBand network
  - Mellanox ConnectX5 or ConnectX6
  - OFED 4.6-1.0.1.1 or higher
    - For GPUDirect Storage: install with --upstream-libs and --dpdk

Note: GPUDirect Storage completely bypasses the kernel and does not utilize the page cache. Standard RDMA clients still utilize the page cache.

Note: RDMA/GPUDirect Storage technology is not supported when working with a cluster with mixed IB and Ethernet networking.

Running **`weka cluster nodes`** will indicate if the RDMA is utilized, for example:

```
# weka cluster nodes
NODE ID      HOST  ID     ROLES        NETWORK
NodeId: 0    HostId: 0    MANAGEMENT   UDP
NodeId: 1    HostId: 0    FRONTEND     DPDK / RDMA
NodeId: 2    HostId: 0    COMPUTE      DPDK / RDMA
NodeId: 3    HostId: 0    COMPUTE      DPDK / RDMA
NodeId: 4    HostId: 0    COMPUTE      DPDK / RDMA
NodeId: 5    HostId: 0    COMPUTE      DPDK / RDMA
NodeId: 6    HostId: 0    DRIVES       DPDK / RDMA
NodeId: 7    HostId: 0    DRIVES       DPDK / RDMA
```

Note: GPUDirect Storage is auto-enabled and detected by the system. To enable/disable RDMA networking altogether on the cluster or a specific client, contact the customer support team.

Chapter 3: Managing the system using the GUI

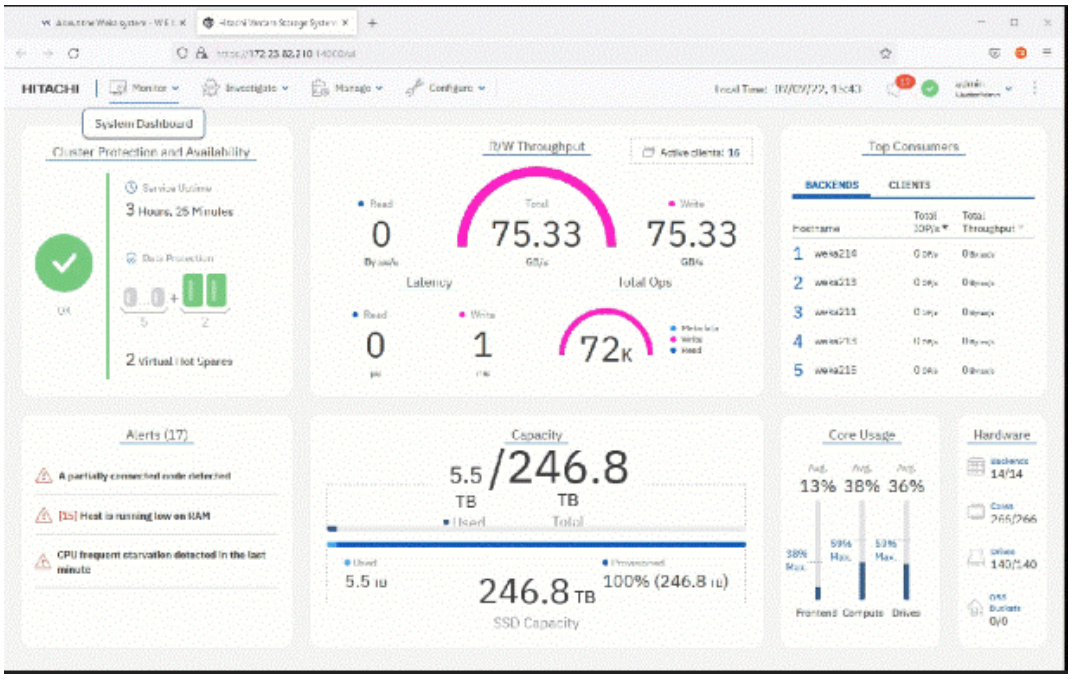
The Content Software for File GUI application enables you to configure, administer, and monitor the Content Software for File system. This page provides an overview of the primary operations, access to the GUI, and system dashboard.

GUI overview

The Content Software for File GUI application is the administration tool for your Content Software for File system. Use this tool for system configuration, filesystems management, user management, and investigation of alarms, events, and statistics.

Content Software for File GUI application supports the following functions:

- Configuration:
  - Configure the cluster, such as data availability, license, security, and central monitoring.
  - Configure the backend servers and expose the data in different protocols.
  - Manage local users and set up the user directory.
  - Create and manage organizations and their quotas.
- Management:
  - Manage the filesystems, including tiering, thin provisioning, and encryption.
  - Manage snapshots.
  - Manage the object store buckets.
  - Manage the filesystem protocols: SMB, S3, and NFS.
- Investigation:
  - Investigate events
  - Investigate overtime statistics, such as total operations, R/W throughput, CPU usage, and read or write latency.
- Monitoring:
  - View the cluster protection and availability.
  - View the R/W throughput
  - View the backend and client top consumers.
  - View alarms. View the used, provisioned, and total capacity.
  - View the frontend, compute, and drive cores usage.
  - View the hardware components (active/total).



Accessing the Content Software for File GUI

The Content Software for File GUI is a web application that you can access using an already configured account and has the appropriate rights to configure, administer, or view.

You can access the Content Software for File GUI with any standard browser using the address: `https://<weka system or host name>:14000`

For example:

`https://WekaProd:14000` or `https://weka01:14000`.

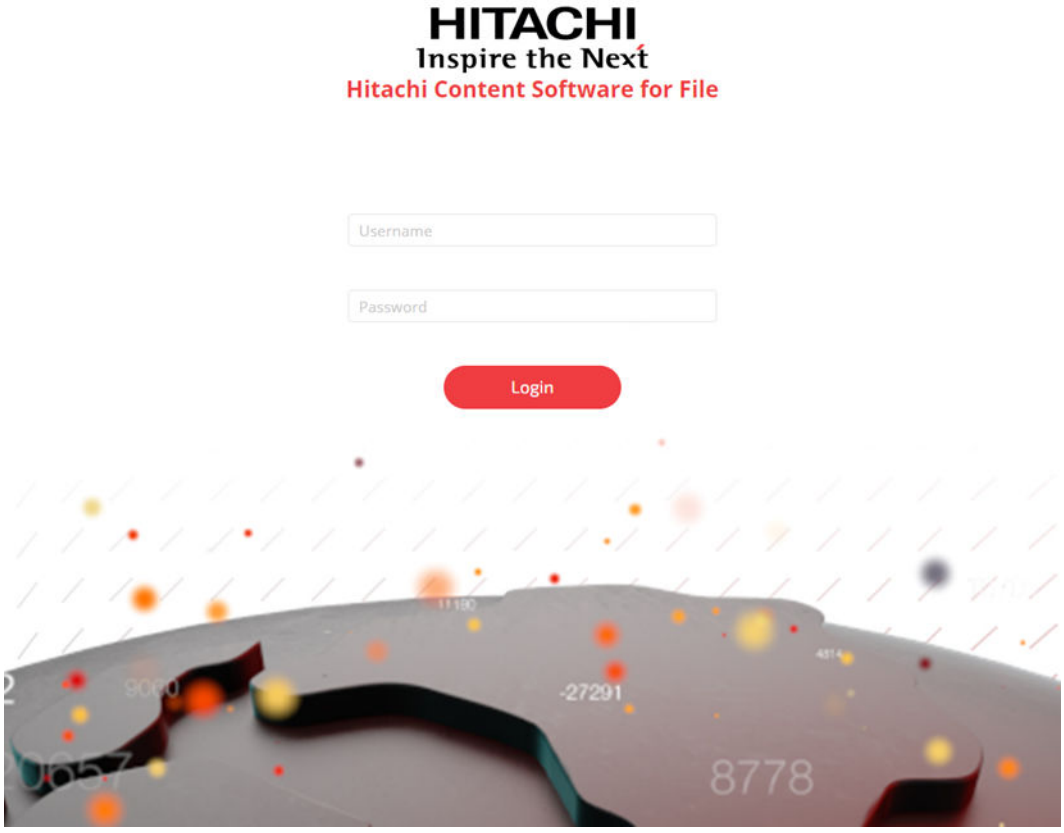
Note: On AWS installations, you can access the Content Software for File GUI from the self-service portal. In the Outputs tab of the CloudFormation stack, click the GUI link.

Before you begin

Make sure that port 14000 is open in the firewall of your organization.

Procedure

1. In your browser, go to `https://<weka system or host name>:14000`. The sign-in page opens.



2. Sign in with the username and password of an account with cluster administration or organization administration privileges. For details about the account types, see User management in the related topics.

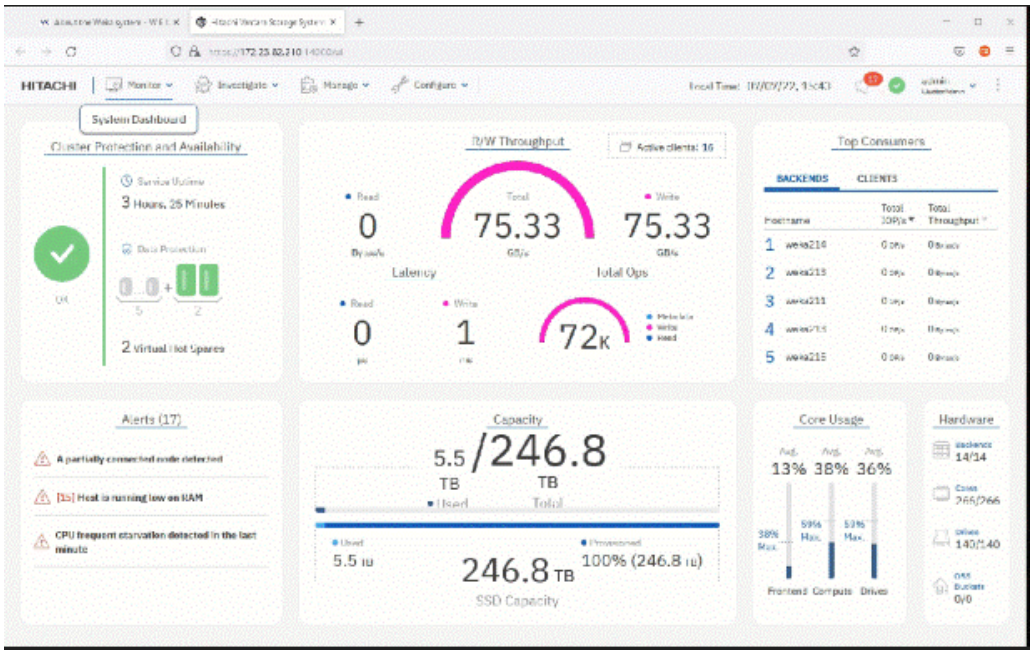
   The system dashboard opens.

   Note: The initial default username and password are admin and admin. In the first sign-in, Content Software for File GUI enforces changing the admin password.

System Dashboard

The system dashboard contains widgets that provide an overview of the Content Software for File system, including an overall status, R/W throughput, top consumers, alerts, capacity, core usage, and hardware.

The system dashboard opens by default when you sign in. If you select another menu and you want to display the dashboard again, select Monitor > System Dashboard, or click the Hitachi logo.

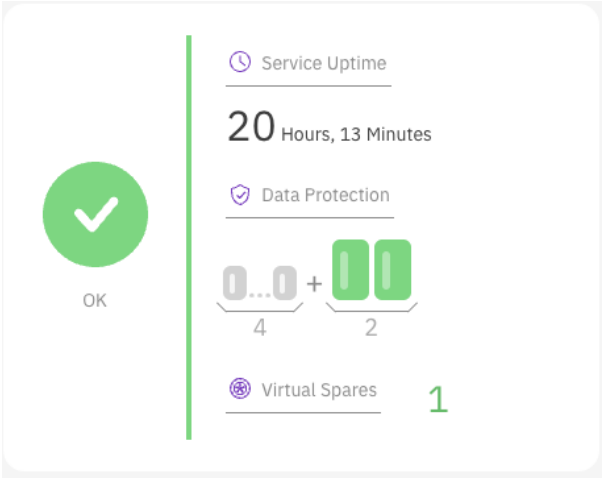


Cluster Protection and Availability widget

This widget shows the overall status of the system's health and protection.

The overall status widget includes the following indications:

- Service Uptime: The elapsed time since the I/O services started.
- Data Protection: The number of data drives and protection parity drives. The color of the protection parity drives indicates their status.
- Virtual (Hot) Spares: The number of failure domains that the system can lose and still complete the data rebuild while maintaining the same net capacity.

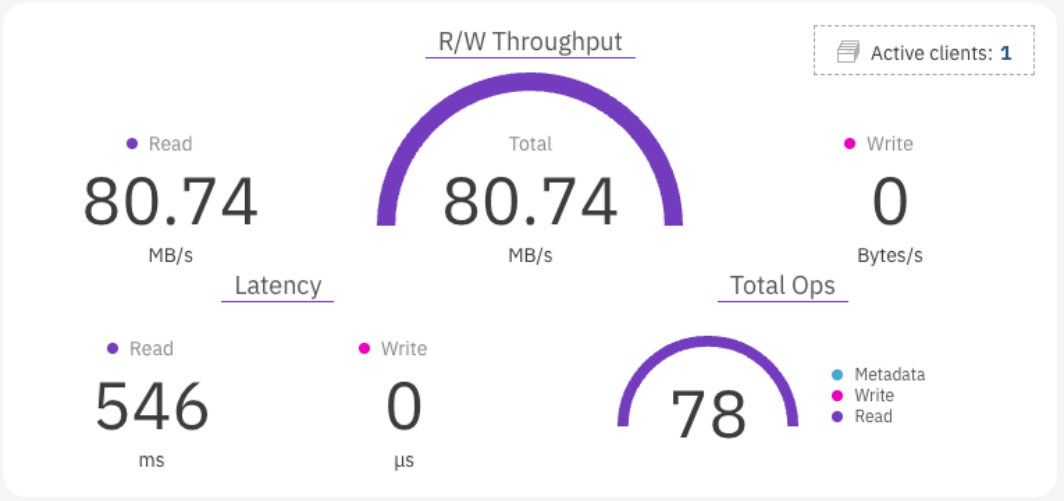


R/W Throughput widget

This widget shows the current performance statistics aggregated across the cluster.

The R/W Throughput widget includes the following indications:

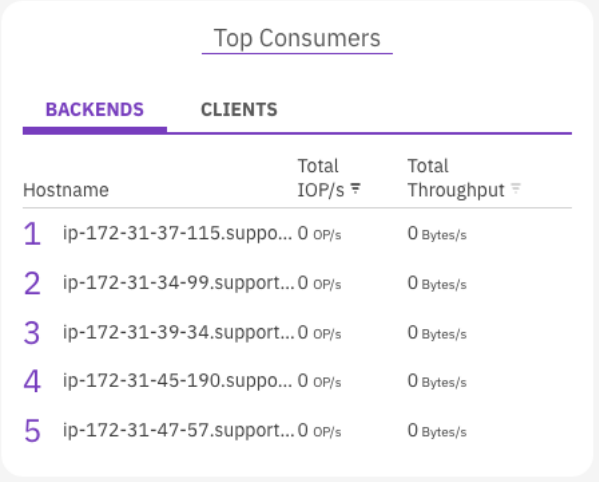
- Throughput: The total throughput.
- Total Ops: The number of cluster operations.
- Latency: The average latency of R/W operations.
- Active clients: The number of clients connected to the cluster.



Note: Selecting the titles R/W Throughput, Latency, and Total Ops displays the statistics page. Selecting the title Active clients displays the client machines page.

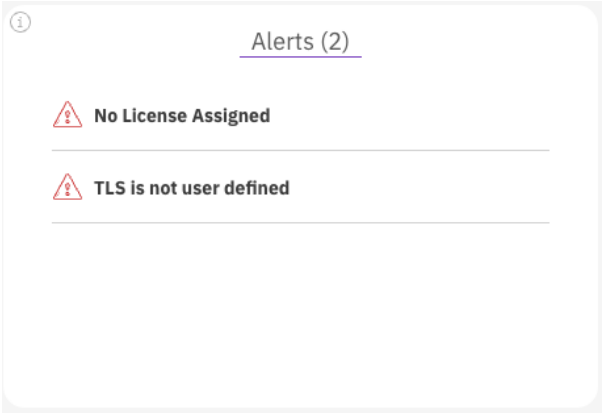
Top Consumer widget

This widget shows the backend and client hosts in the system. You can sort the list of hosts by total IO operations per second or by total throughput.



Alerts widget

This widget shows the alerts that are not muted.

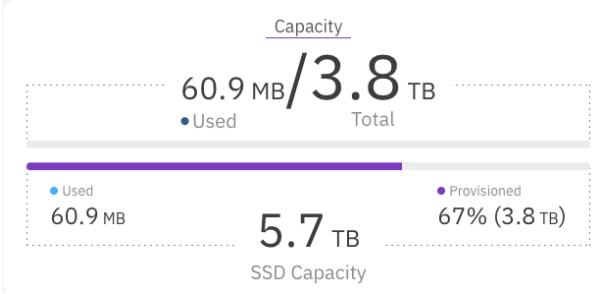


Capacity widget

This widget shows an overview of the managed capacity.

The top bar indicates the total capacity provisioned for all filesystems and the used capacity. For tiered filesystems, the total capacity also includes the Object Store part.

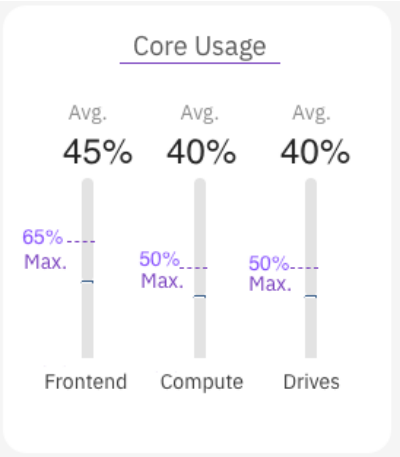
The bottom bar indicates the total SSD capacity available in the system, the provisioned capacity, and the used capacity.



Note: Selecting the title Capacity displays the filesystems page.

Core Usage widget

This widget shows the average usage and the maximum load level of the Frontend, Compute, and Drive cores.



Hardware widget

This widget shows an overview of the hardware components (active/total).

The hardware components include:

- Backends: The number of the servers.
- Cores: The number of cores configured for running processes in the backends.
- Drives: The number of drives.
- OBS Buckets: The number of the object-store buckets.

Note: Selecting the titles Backends, Cores and Drives displays the backend machines page. Selecting the title OBS buckets displays the object store buckets page.

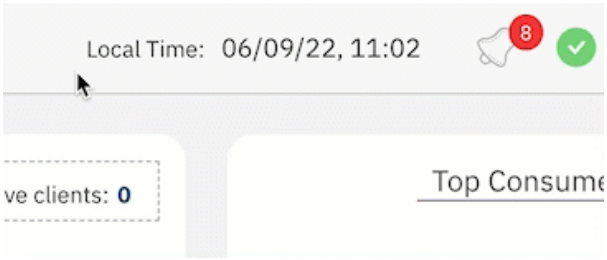
Switch the display time

Timestamps in events and statistics are logged internally in UTC. Weka GUI displays the timestamps in local or system time. You can switch between the local and system time.

Switching the display time may be required when the customer, Weka support, and the Weka system are in different time zones. In this situation, the customer and Weka support can switch the display to system time instead of local time, so both view the identical timestamps.

Procedure

1. On the top bar, point to the timestamp.
2. Depending on the displayed time, select **Switch to System Time** or **Switch to Local** Time.



Chapter 4: Managing filesystems and object stores

The details on how to view and manage filesystems, object stores, and filesystem groups using the GUI are provided.

Overview

The management of object stores, filesystem groups and filesystems is an integral part of the successful running and performance of the Content Software for File system and overall data lifecycle management.

The pages in this section cover the following subjects:

- Managing object stores (on page 52).
- Managing filesystem groups (on page 56).
- Managing filesystems (on page 59).
- Attaching or detaching object stores to or from filesystems (on page 63).

Managing object stores

Using the GUI, you can perform the following actions:

- Editing default object stores using the GUI (on page 52)
- Viewing object stores using the GUI (on page 53)
- Adding an object store using the GUI (on page 53)
- Editing an object store using the GUI (on page 55)
- Deleting an object store using the GUI (on page 56)

Editing default object stores using the GUI

Object store buckets can reside in different physical object stores. To achieve good QoS between the buckets, Weka requires to map the buckets to the physical object store.

You can edit the default local and remote object stores to meet your connection demands. When you add an object store bucket, you apply the relevant object store on the bucket.

Editing the default object store provides you with the following additional advantages:

- Set restrictions on downloads from a remote object store. For on-premises systems where the remote bucket is in the cloud, to reduce the cost, you set a very low bandwidth for downloading from a remote bucket.
- Ease of adding new buckets. You can set the connection parameters on the object store level and, if not specified differently, automatically use the default settings for the buckets you add.

Procedure

1. From the menu, select **Manage > Object Stores**.
2. On the left, select the pencil icon near the default object store you want to edit
3. On the **Edit Object Store** dialog, set the following:
   - **Type**: Select the type of object store.
   - **Buckets Default Parameters**: Set the protocol, hostname, port, bucket folder, authentication method, region name, access key, and secret key.

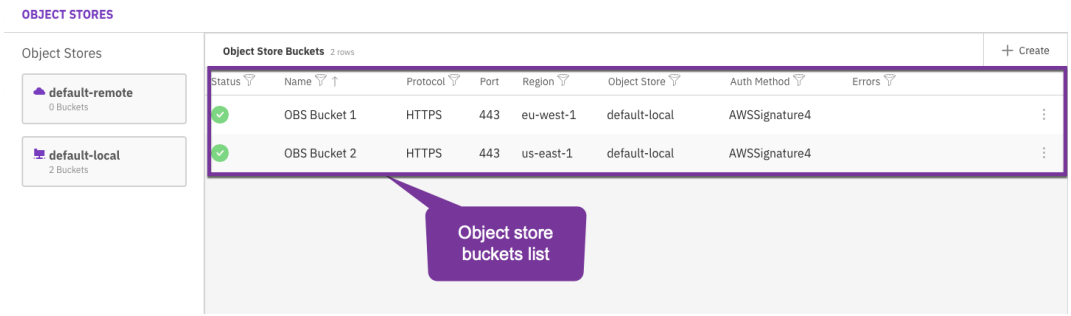
> **Note:** If using the AWS object store type and access from the Weka EC2 instances to the object store is granted by the IAM roles, it is not mandatory to set the access and secret keys in the Edit Object Store dialog.

Viewing object stores using the GUI

The object store buckets are displayed on the Object Stores page. Each object store indicates the status, bucket name, protocol (HTTP/HTTPS), port, region, object store location (local or remote), authentication method, and error information (if exists).

From the menu, select Manage > Object Stores.

The following example shows two object store buckets.

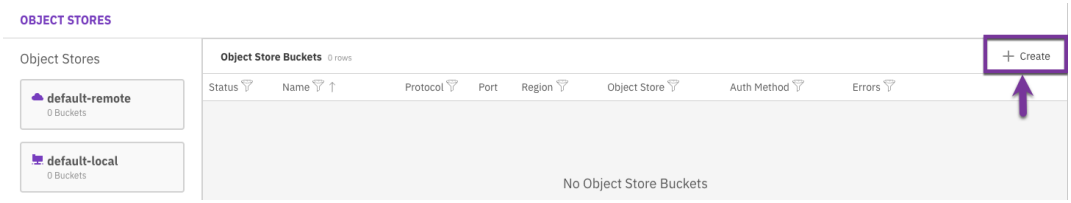


Adding an object store using the GUI

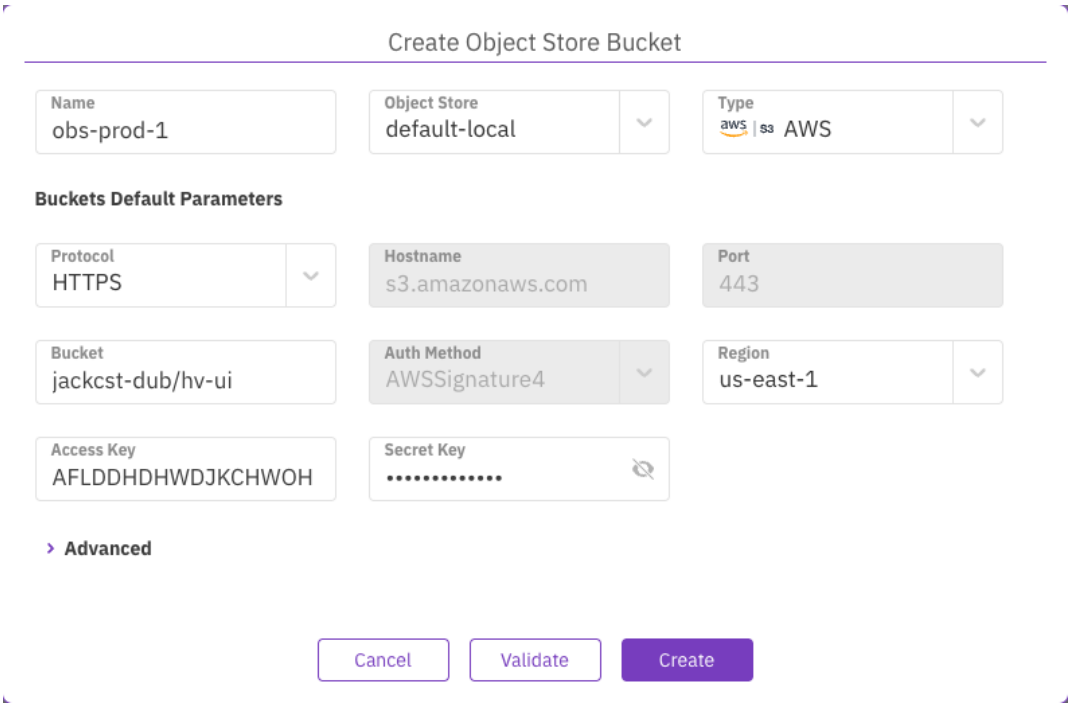
Add object store buckets to be used for tiering or snapshots.

Procedure

1. From the menu, select **Manage > Object Stores**.
2. Select the **+Create** button.

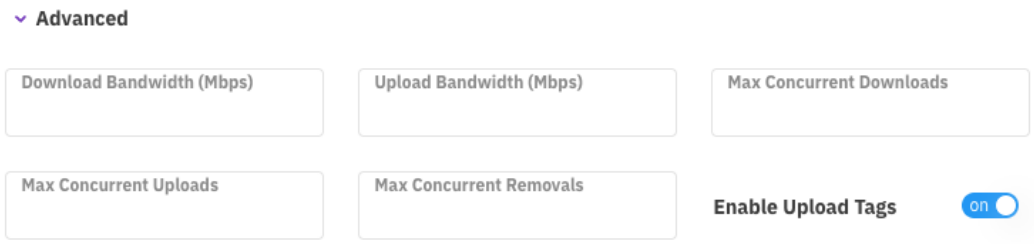


3. In the **Create Object Store Bucket** dialog, set the following:
   - **Name**: Enter a meaningful name for the bucket.
   - **Object Store**: Select the location of the object store. For tiering and snapshots, select the local object store. For snapshots only, select the remote object store.
   - **Type**: Select the type of object store.
   - **Buckets Default Parameters**: Set the protocol, hostname, port, bucket folder, authentication method, region name, access key, and secret key.



4. To validate the connection to the object store bucket, select **Validate**.
5. Optional: If your deployment requires a specific upload and download configuration, select Advanced, and set the parameters:
   - **Download Bandwidth**: Object store download bandwidth limitation per core (Mbps).
   - **Upload Bandwidth**: Object store upload bandwidth limitation per core (Mbps).
   - **Max concurrent Downloads**: Maximum number of downloads concurrently performed on this object store in a single IO node.
   - **Max concurrent Uploads**: Maximum number of uploads concurrently performed on this object store in a single IO node.

- **Max concurrent Removals**: Maximum number of removals concurrently performed on this object store in a single IO node,
- **Enable Upload Tags**: Whether to enable object-tagging or not.



6. Select **Create**.

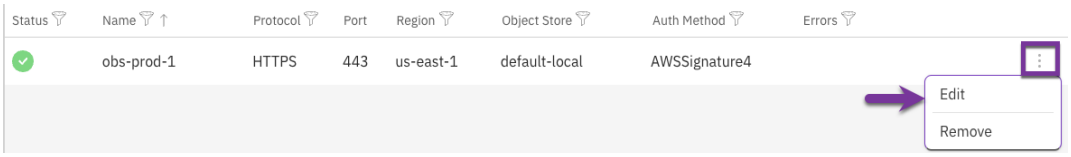
> **Note:** If an error message about the object store bucket configuration appears, to save the configuration, select Create Anyway.

Editing an object store using the GUI

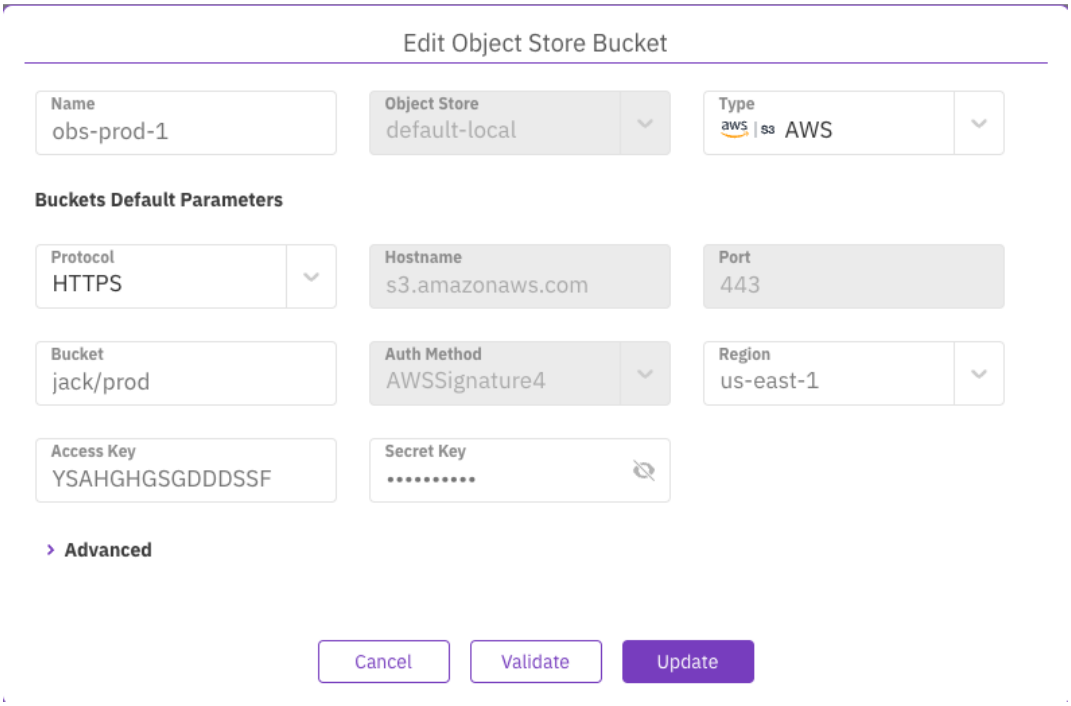
You can modify the object store bucket parameters according to your demand changes over time.

Procedure

1. From the menu, select **Manage > Object Stores**.
2. Select the three dots on the right of the object store you want to modify, and select **Edit**.



3. In the Edit Object Store Bucket dialog, modify the details, and select **Update**.



Deleting an object store using the GUI

You can delete an object store bucket if it is no longer required. The data in the object store remains intact.

Procedure

1. From the menu, select **Manage > Object Stores**.
2. Select the three dots on the right of the object store bucket you want to delete, and select **Remove**.
3. To confirm the object store bucket deletion, select **Yes**.

Managing filesystem groups

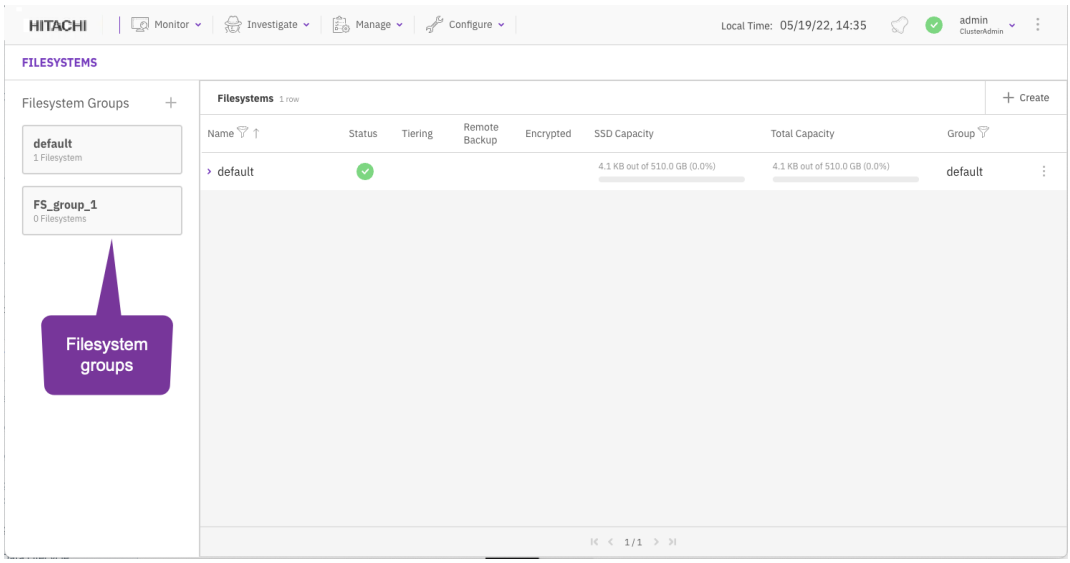
A filesystem group defines the policy of the drive retention period and the tiering cue time. The Content Software for File system can include up to eight filesystem groups.

Viewing filesystem groups using the GUI

The filesystem groups are displayed on the Filesystems page. Each filesystem group indicates the number of filesystems that use it.

From the menu, select Manage > Filesystems.

The following example shows two filesystem groups defined in the system.



Adding a filesystem group using the GUI

Adding a filesystem group is required when adding a filesystem. If you want to apply a different tiering policy on specific filesystems, you can create more file system groups.

Procedure

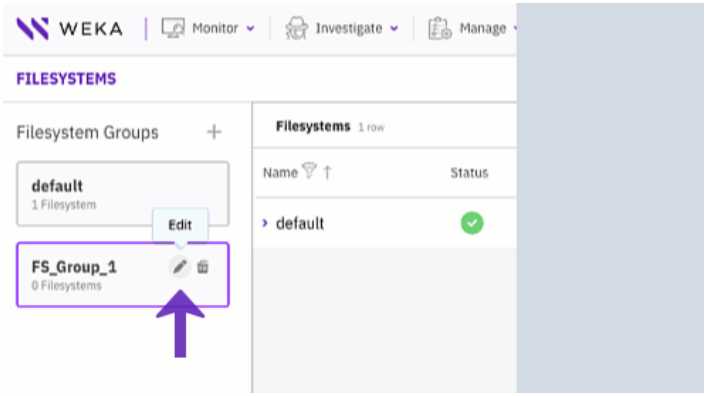
1. From the menu, select **Manage > Filesystems**.
2. Select the **+** sign right to the Filesystem Groups title.
3. In the **Create Filesystem Group** dialog, set the following:
   - **Name**: Enter a meaningful name for the filesystem group.
   - **Drive Retention Period**: Set the number of days to keep the data on the SSD before it is copied to the object store. After this period, the copy of the data is deleted from the SSD.
   - **Tiering Cue**: Set the time to wait after the last update, before the data is copied from the SSD and sent to the object store.
4. Select **Create**.

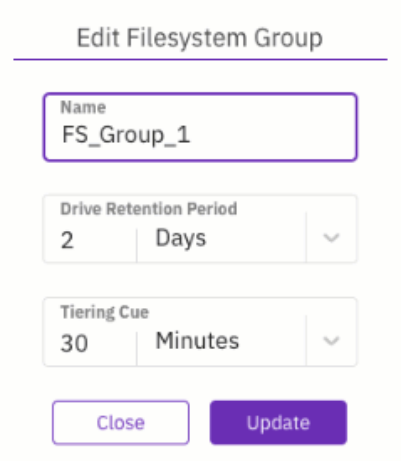
Editing a filesystem group using the GUI

You can edit the filesystem group policy according to your system requirements.

Procedure

1. From the menu, select **Manage > Filesystems**.
2. Select the filesystem group you want to edit.
3. Select the pencil sign right to the filesystem group name.
4. In the **Edit Filesystem Group** dialog, update the settings as you need. (See the parameter descriptions in the Add a Filesystem Group topic.





im

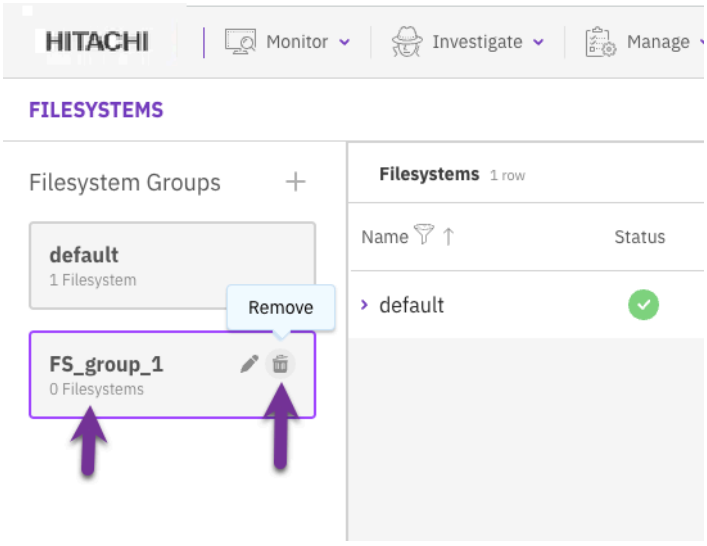
5. Select **Update**.

Deleting a filesystem group using the GUI

You can delete a filesystem group that is no longer used by any filesystem.

Procedure

1. From the menu, select **Manage > Filesystems**.
2. Select the filesystem group you want to delete.
3. Verify that the filesystem group is not used by any filesystems (indicates 0 filesystems).



4. Select the **Remove** icon. In the pop-up message, select **Yes** to delete the filesystem group.

Managing filesystems

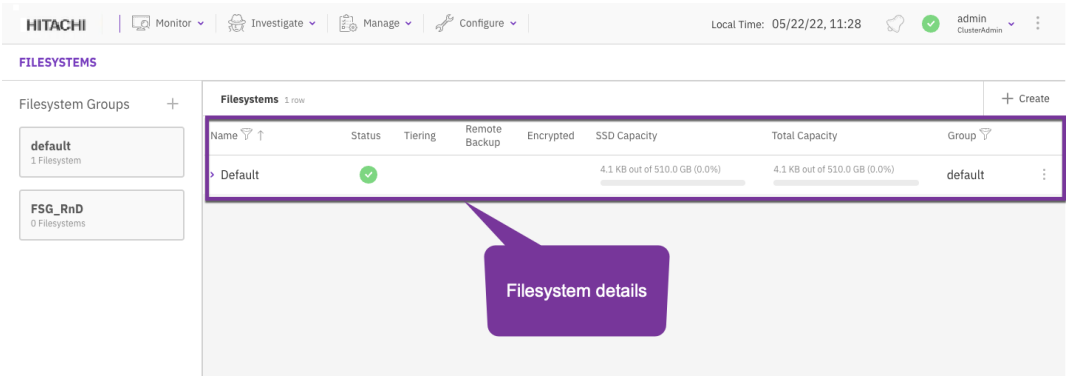
How to view and manage filesystem groups using the GUI.

- Viewing filesystem groups using the GUI (on page 56)
- Adding a filesystem using the GUI (on page 60)
- Editing a filesystem using the GUI (on page 62)
- Deleting a filesystem using the GUI (on page 62)

Viewing filesystems using the GUI

The filesystems are displayed on the Filesystems page. Each filesystem indicates the status, tiering status, backup status, encryption status, SDD capacity, total capacity, and the filesystem group used.

From the menu, select Manage > Filesystems.



Adding a filesystem using the GUI

When creating a Weka system on-premises, it does not contain any filesystem. You need to add it and set its properties, such as capacity, group, tiering, thin provisioning, and encryption.

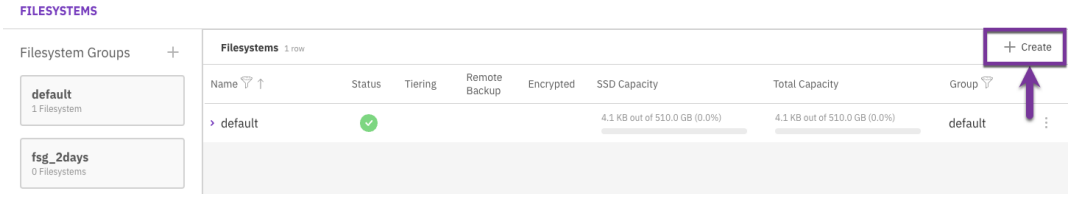
When creating a Weka system in AWS using the cloud formation, the Weka system contains a default filesystem, which is provisioned with the maximum capacity. If your deployment requires more filesystems with different settings, first reduce the provisioned capacity of the default filesystem, and then add a filesystem with the properties that meet your specific needs.

Before you begin

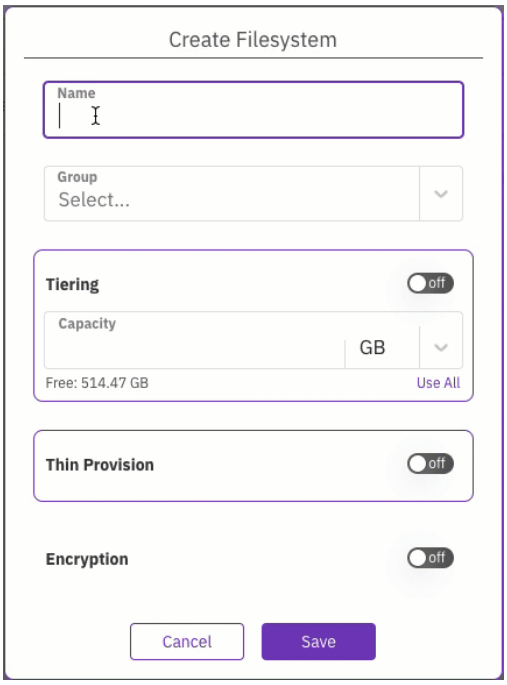
- Verify that the system has free capacity.
- Verify that a filesystem group is already set.
- If tiering is required, verify that an object store bucket is set.
- If encryption is required, verify that a KMS is configured.

Procedure

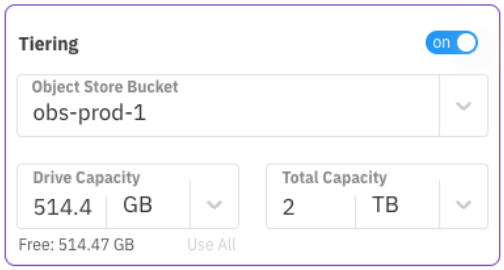
1. From the menu, select **Manage > Filesystems**.
2. Select the **+Create** button.



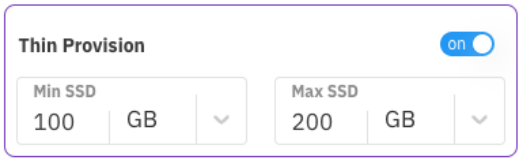
3. In the **Create Filesystem** dialog, set the following:
   - **Name**: Enter a meaningful name for the filesystem.
   - **Group**: Select the filesystem group that fits your filesystem.
   - **Capacity**: Enter the storage size to provision, or select **Use All** to provision all the free capacity.



4. Optional: If Tiering is required and an object store bucket is already defined, select the toggle button, and set the details of the object store bucket:
   - **Object Store Bucket**: Select a predefined object store bucket from the list.
   - **Drive Capacity**: Enter the capacity to provision on the SSD, or select Use All to use all free capacity.
   - **Total Capacity**: Enter the total capacity of the object store bucket, including the drive capacity.



5. Optional: If Thin Provision is required, select the toggle button, and set the minimum and the maximum capacity of the SSD to use for thin provisioning.



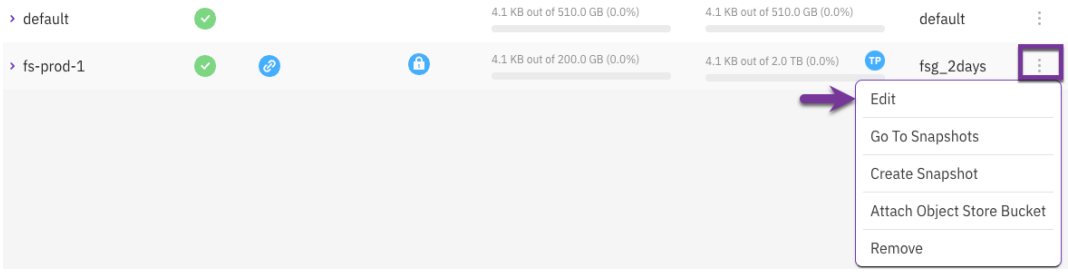
6. Optional: If Encryption is required and your Weka system is deployed with a KMS, select the toggle button.
7. Select **Save**.

Editing a filesystem using the GUI

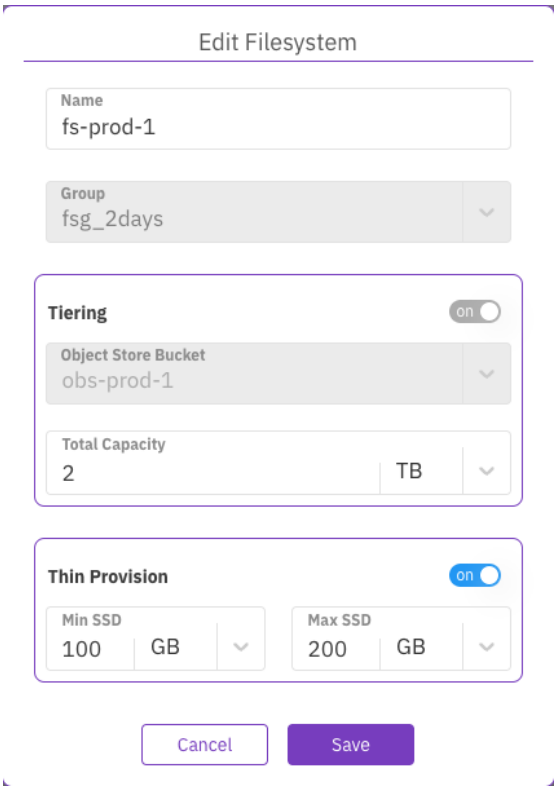
You can modify the filesystem parameters according to your demand changes over time. The parameters that you can modify include, filesystem name, capacity, tiering, and thin provisioning (but not encryption).

Procedure

1. From the menu, select **Manage > Filesystems**.
2. Select the three dots on the right of the filesystem you want to modify, and select **Edit**.



3. In the **Edit Filesystem** dialog, modify the parameters according to your requirements. (See the parameter descriptions in the Add a filesystem topic.



4. Select **Save**.

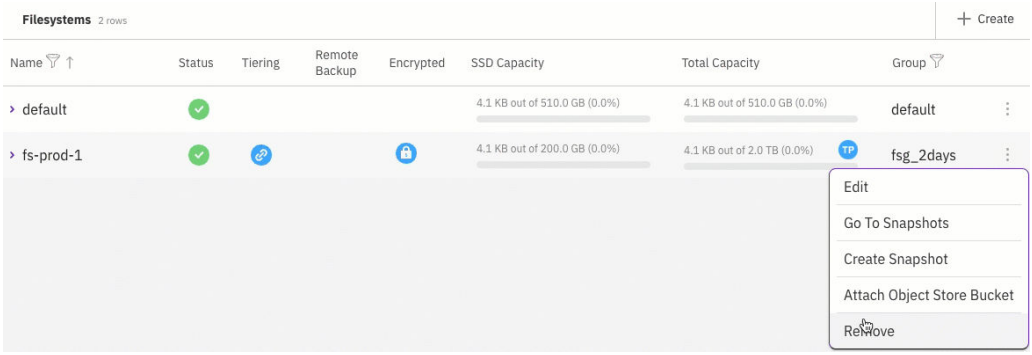
Deleting a filesystem using the GUI

You can delete a filesystem if its data is no longer required. Deleting a filesystem does not delete the data in the tiered object store bucket.

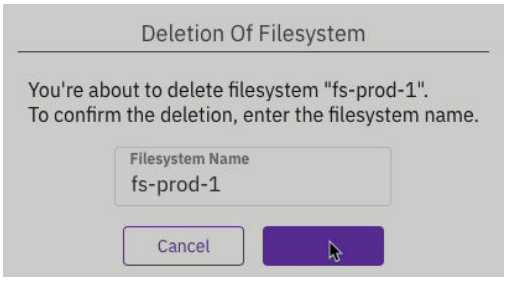
Note: If you need to delete also the data in the tiered object store bucket, see Delete a Filesystem in the CLI section.

Procedure

1. From the menu, select **Manage > Filesystems**.
2. Select the three dots on the right of the filesystem you want to delete, and select **Remove**.



3. To confirm the filesystem deletion, enter the filesystem name and select **Confirm**.



Attaching or detaching object stores to or from filesystems

Attachment of a local object store bucket to a filesystem

Two local object store buckets can be attached to a filesystem, but only one of the buckets is writable. A local object store bucket is used for both tiering and snapshots. When attaching a new object store bucket to an already tiered filesystem, the existing object-store bucket becomes read-only, and the new object store bucket is read/write. Multiple object stores allow a range of use cases, including migration to different object stores, scaling of object store capacity, and increasing the total tiering capacity of filesystems.

When attaching an object store bucket to a non-tiered filesystem, the filesystem becomes tiered.

Detachment of a local object store bucket from a filesystem

Detaching a local object-store bucket from a filesystem migrates the filesystem data residing in the object store bucket either to the writable object store bucket (if one exists) or to the SSD.

When detaching, the background task of detaching the object-store bucket begins. Detaching can be a long process, depending on the amount of data and the load on the object stores.

Note: Detaching an object store bucket is irreversible. Attaching the same bucket again is considered as re-attaching a new bucket regardless of the data stored in the bucket.

Attachment of a local object store bucket to a filesystem

Two local object store buckets can be attached to a filesystem, but only one of the buckets is writable. A local object store bucket is used for both tiering and snapshots. When attaching a new object store bucket to an already tiered filesystem, the existing object-store bucket becomes read-only, and the new object store bucket is read/write. Multiple object stores allow a range of use cases, including migration to different object stores, scaling of object store capacity, and increasing the total tiering capacity of filesystems.

When attaching an object store bucket to a non-tiered filesystem, the filesystem becomes tiered.

Detachment of a local object store bucket from a filesystem

Detaching a local object-store bucket from a filesystem migrates the filesystem data residing in the object store bucket either to the writable object store bucket (if one exists) or to the SSD.

When detaching, the background task of detaching the object-store bucket begins. Detaching can be a long process, depending on the amount of data and the load on the object stores.

Note: Detaching an object store bucket is irreversible. Attaching the same bucket again is considered as re-attaching a new bucket regardless of the data stored in the bucket.

Migration to a different object store

When detaching from a filesystem tiered to two object store buckets, only the read-only object-store bucket can be detached. In such cases, the background task copies the relevant data to the writable object store.

Un-tiering a filesystem

Detaching from a filesystem tiered to one object store bucket un-tiers the filesystem and copies the data back to the SSD.

Note: The SSD must have sufficient capacity. That is, the allocated SSD capacity must be at least the total capacity used by the filesystem.

On completion of detaching, the object-store bucket does not appear under the filesystem when using the weka fs command. However, it still appears under the object store and can be removed if it is not being used by any other filesystem. The data in the read-only object-store bucket remains in the object-store bucket for backup purposes. If this is unnecessary or the reclamation of object store space is required, it is possible to delete the object-store bucket.

Note: Before deleting an object-store bucket, remember to take into account data from another filesystem or data not relevant to the Weka system on the object-store bucket.

Note: After the migration process is done, while relevant data is migrated, old snapshots (and old locators) reside on the old object-store bucket. To recreate snapshots locators on the new object-store bucket, snapshots should be re-uploaded to the (new) object-store bucket.

Migration considerations

When migrating data (using the detach operation) you would like to copy only the necessary data (to reduce migration time and capacity), however, you may want to keep snapshots in the old object-store bucket.

Migration workflow

The order of the following steps is important.

Procedure

1. Attach a new object store bucket (the old object store bucket becomes read-only).
2. Delete any snapshot that does not need to be migrated. This action keeps the snapshot on the old bucket but does not migrate its data to the new bucket.
3. Detach the old object store bucket.

Note: If you perform the workflow steps in a different order, the snapshots can be completely deleted from any of the object store buckets. It is also possible that the snapshots are already in a migration process, and cannot be deleted until the migration is completed.

Attaching a remote object store bucket

One remote object-store bucket can be attached to a filesystem. A remote object store bucket is used for backup only, and only snapshots are uploaded to it using Snap-To-Object. The snapshot uploads are incremental to the previous one.

Detaching a remote object store bucket

Detaching a remote object-store bucket from a filesystem keeps the backup data within the bucket intact. It is still possible to use these snapshots for recovery.

Attaching an object store bucket to a filesystem using the GUI

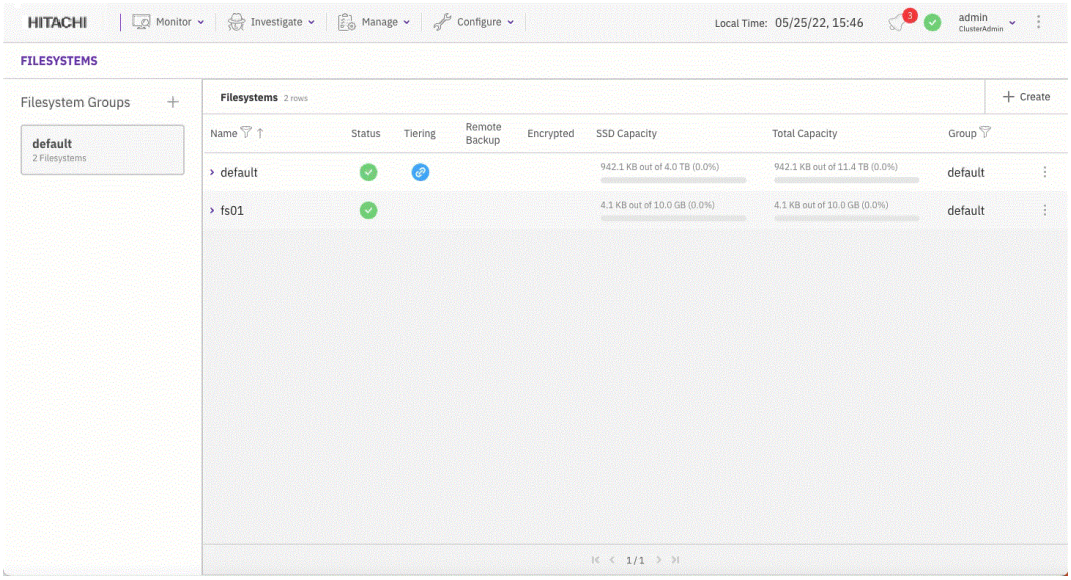
Before you begin

Verify that an object store bucket is available.

One remote object-store bucket can be attached to a filesystem. A remote object store bucket is used for backup only, and only snapshots are uploaded to it using Snap-To-Object. The snapshot uploads are incremental to the previous one.

Procedure

1. From the menu, select **Manage > Filesystems**.
2. On the **Filesystem** page, select the three dots on the right of the filesystem that you want to attach to the object store bucket. Then, from the menu, select **Attach Object Store Bucket**.
3. On the Attach Object Store Bucket dialog, select the relevant object store bucket.

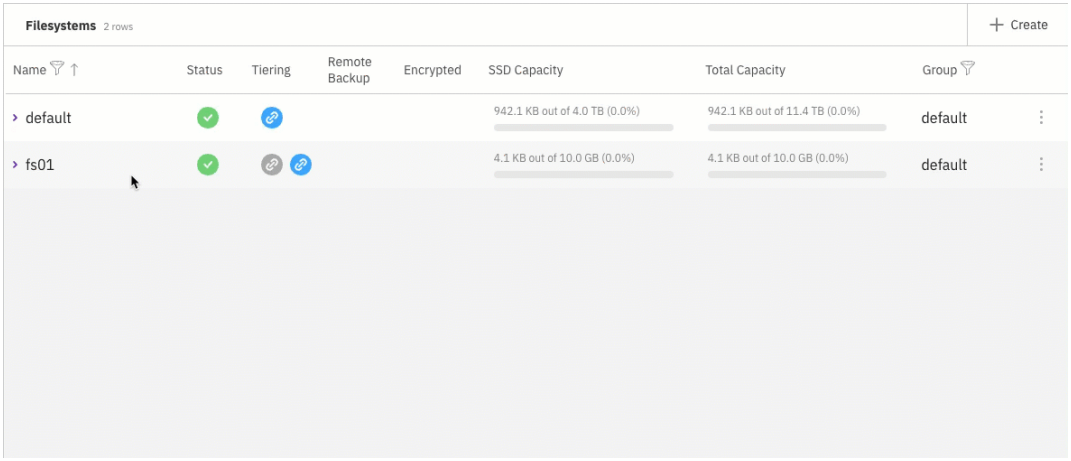


Detaching an object store bucket from a filesystem using the GUI

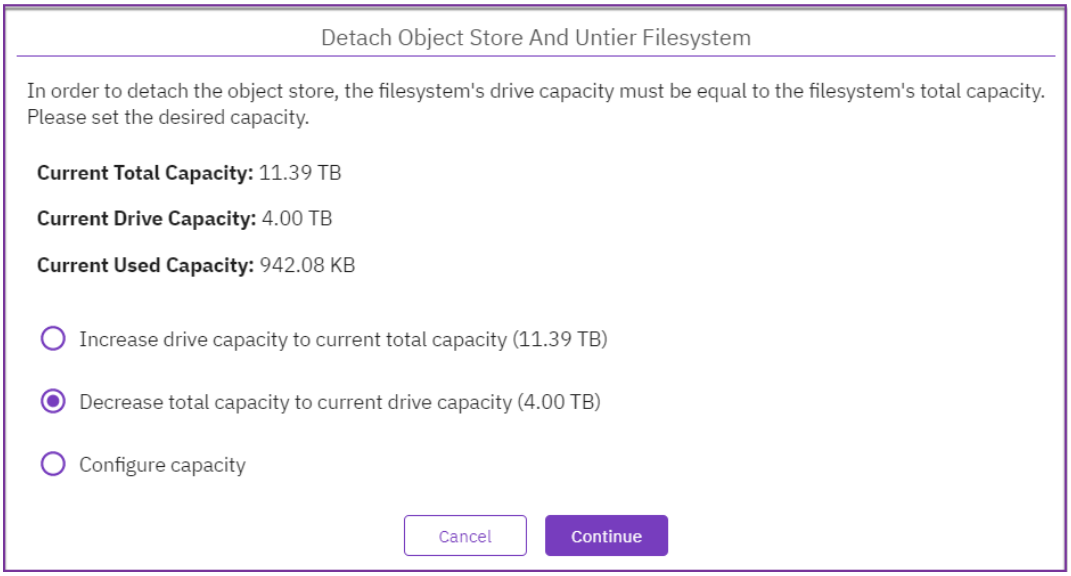
Detaching a local object store bucket from a filesystem migrates the filesystem data residing in the object store bucket either to the writable object store bucket (if one exists) or to the SSD.

Procedure

1. From the menu, select **Manage > Filesystems**.
2. On the **Filesystem** page, select the filesystem from which you want to detach the object store bucket.
3. From the **Detach Object Store Bucket** dialog, select **Detach**. If the filesystem is attached to two object store buckets (one is read-only and the other is writable), you can detach only the one that is read-only. The data of the detached object store bucket is migrated to the writable object store bucket.
4. In the message that appears, to confirm the detachment, select **Yes**.



5. If the filesystem is tiered and only one object store is attached, detaching the object store bucket opens the following message:



6. Object store buckets usually expand the filesystem capacity. Un-tiering of a filesystem requires adjustment of its total capacity. Select one of the following options:
   - Increase the SSD capacity to match the current total capacity.
   - Reduce the total filesystem capacity to match the SSD capacity or the used capacity (the decrease option depends on the used capacity).
   - Configure a different capacity.

   Note: Used capacity must be taken into account. Un-tiering takes time to propagate the data from the object store to the SSD. When un-tiering an active filesystem, to accommodate the additional writes during the detaching process, it is recommended to adjust to a higher value than the used capacity.

7. Select the option that best meets your needs, and select **Continue**.
8. In the message that appears, select **Detach** to confirm the action.

Chapter 5: Advanced data lifecycle management

This section explains how data lifecycle is maintained when working with a tiered Content Software for File system configuration together with the options for control. The following subjects are covered:

- Advanced explanation of time-based policies (on page 68) for data storage location.
- System behavior when tiering, accessing or deleting data in tiered filesystems (on page 73).
- Transition between SSD-only and tiered filesystems (on page 76).
- For how to manually pre-fetch tiered data from an object-store and release of data from SSD to the object-store, see *Content Software for File Command Line Reference Guide, Manual fetch and release of data* section.

Advanced time-based policies for data storage location

This page provides an in-depth explanation on Advanced data lifecycle management (on page 68) overview section.

Data retention period policy

The Drive Retention Period policy refers to the amount of time you want to keep a copy of the data on SSD that you previously offloaded/copied to the object storage via the Tiering Cue Policy described further below.

Consider a scenario of a 100 TB filesystem (total capacity), with 100 TB of SSD space (as explained in The Role of SSDs in tiered Content Software for File configurations (on page 32) section). If the data Drive Retention Period policy is defined as 1 month and only 10 TB of data are written per month, it will probably be possible to maintain data from the last 10 months on the SSDs. On the other hand, if 200 TB of data is written per month, it will only be possible to maintain data from half of the month on the SSDs. Additionally, there is no guarantee that the data on the SSDs is the data written in the last 2 weeks of the month, which also depends on the Tiering Cue.

To further help describe this section, let us use an example where the Tiering Cue Policy described below is set to 1 day, and the Drive Retention Period is set to 3 days. After one day, the Weka system offloads period 0's data to the object store. Setting the Drive Retention Period to 3 days means leaving a copy of that data in Weka Cache for three days, and after three days, it is removed from the Weka Cache. The data is not gone, it is on the object store, and if an application or a user accesses that data, it is pulled back from the object store and placed back on the Weka SSD tier where it is tagged again with a new Tiering Cue Policy Period.

Consequently, the data Retention Period policy determines the resolution of the Content Software for File system release decisions. If it is set to 1 month and the SSD capacity is sufficient for 10 months of writing, then the first month will be kept on the SSDs.

Note: If the Content Software for File 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 Content Software for File 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.

Note: If the data writing rate is always high and the Content Software for File system fails to successfully release the data to the object store, an Object Store Bottleneck will occur. If the bottleneck continues, this will also result in a Policy Violation event.

Tiering cue policy

The *Tiering Cue* defines the period of time to wait before the data is copied from the SSD and sent to the object store. It is typically used when it is expected that some of the data being written will be rewritten/modified/deleted in the short term.

The Content Software for File system integrates a rolling progress control with three rotating periods of 0, 1, and 2.

1. Period 0: All data written is tagged as written in the current period.
2. Period 1: The switch from 0 to 1 is according to the Tiering Cue policy.
3. Period 2: Starts after the period of time defined in the Tiering Cue, triggering the transfer of data written in period 0 from the SSD to the object-store.

Note: Not all data is transferred to the object store in the order that it was written. If, for example, the Tiering Cue is set to 1 month, there is no priority or order in which the data from the whole month is released to the object store; data written at the end of the month may be released to the object store before data written at the beginning of the month.

For example: Let us say your Tiering Cue Policy is set to 1 day. All data written within the first day is tagged for Period 0. After one day, and for the next day, the next set of data is tagged for Period 1, and the data written in the next day is tagged for Period 2. As Period 0 rolls around to be next, the data marked for Period 0 is then offloaded to the object store, and new data is then tagged for Period 0. When Period 1 rolls around to be next, it is time to offload the data tagged for Period 1 to the object store and so on.

One important caveat to mention is that in the above example, if none of the data is touched or modified during the time set for the Tiering Cue Policy, then all the data as described will offload to the object store as planned. But let's say there is some data in Period 0 that was updated/modified, that data is pulled out of Period 0 and is then tagged with the current Period of data being written at the moment, let's say that is Period 2. So now, that newly modified data will not get offloaded to the object store until it is Period 2's time. This is true for any data modified residing in one of the 3 Period cycles. It will be removed from its original Period and placed into the current Period that is marking the active writes.

Management of data retention policies

Since the Content Software for File system is a highly scalable data storage system, data storage policies in tiered configurations cannot be based on cluster-wide FIFO methodology, because clusters can contain billions of files. Instead, data retention is managed by time-stamping every piece of data, where the timestamp is based on a resolution of intervals which may extend from minutes to weeks. The Content Software for File system maintains the interval in which each piece of data was created, accessed, or last modified.

Users only specify the data Retention Period and based on this, each interval is one-quarter of the Data Retention Period. Data written, modified, or accessed prior to the last interval is always released, even if SSD space is available.

Note: The timestamp is maintained per piece of data in chunks of up to 1 MB, and not per file. Consequently, different parts of big files may have different tiering states.

For example: In a Content Software for File system that is configured with a Data Retention Period of 20 days, data is split into 7 interval groups, with each group spanning a total of 5 days (5 is 25% of 20, the data Retention Period). If the system starts operating on January 1, then data written, accessed, or modified between January 1-5 is classified as belonging to interval 1, data written, accessed, or modified between January 6-10 belongs to interval 2, and so on. In such a case, the 7 intervals will be timestamped and divided as follows:

Interval	Interval 0	Interval 1	Interval 2	Interval 3	Interval 4	Interval 5	Interval 6
Date span	Jan 1-5	Jan 6-10	Jan 11-15	Jan 16-20	Jan 21-25	Jan 26-30	Jan 31-Feb 4

In the above scenario, there are seven data intervals on the SSDs (the last one is accumulating new/modified data). In addition, another interval is currently being released to the object-store. Yes, the retention period is almost twice as long as the user specifies, as long as there is sufficient space on the SSD. Why? If possible, it provides better performance and reduces unnecessary release/rehydration of data to/from the object-store if data is modified.

Data release process from SSD to object store

At any given moment, the Content Software for File system releases the filesystem data of a single interval, transferring it from the SSD to the object-store. The release process is based on data aging characteristics (as implemented through the intervals system and revolving tags). Consequently, if there is sufficient SSD capacity, only data modified or written before seven intervals will be released. The release process also considers the amount of available SSD capacity through the mechanism of Backpressure. Backpressure works against two watermarks - 90% and 95%. It kicks in when SSD utilization per file system crosses above 95% and stops when it crosses below 90%. It's also important to understand that Backpressure works in parallel and independently of the Tiering Policy. If the SSD utilization crosses the 95% watermark, then data will be released from SSD and sent to the object-store sooner than was configured.

For example: If 3 TB of data is produced every day, that is, 15 TB of data in each interval, the division of data will be as follows:

Interval	Interval 0	Interval 1	Interval 2	Interval 3	Interval 4	Interval 5	Interval 6
Date span	Jan 1-5	Jan 6-10	Jan 11-15	Jan 16-20	Jan 21-25	Jan 26-30	Jan 31-Feb 4
Capacity	15 TB	15 TB	15 TB	15 TB	15 TB	15 TB	15 TB

Now consider a situation where the total capacity of the SSD is 100 TB. The situation in the example above will be as follows:

Interval	Interval 0	Interval 1	Interval 2	Interval 3	Interval 4	Interval 5	Interval 6
Date span	Jan 1-5	Jan 6-10	Jan 11-15	Jan 16-20	Jan 21-25	Jan 26-30	Jan 31-Feb 4
Capacity	15 TB	15 TB	15 TB	15 TB	15 TB	15 TB	15 TB
Overall SSD capacity assigned to the time span between this interval and now	105	90	75	60	45	30	15

Since the resolution in the Content Software for File system is the interval, in the example above the SSD capacity of 100 TB is insufficient for all data written over the defined 35-day Retention Period. Consequently, the oldest, most non-accessed, or modified data, has to be released to the object store. In this example, this release operation will have to be performed in the middle of interval 6 and will involve the release of data from interval 0.

This counting of the age of the data in resolutions of 5 days is performed according to 8 different categories. A constantly rolling calculation, the following will occur in the example above:

- Data from days 1-30 (January 1-30) will all be on the SSD. Some of it may be tiered to the object store, depending on the defined Tiering Cue.
- Data from more than 35 days will be released to the object-store.
- Data from days 31-35 (January 31-February 4) will be partially on the SSD and partially tiered to the object store. However, there is no control over the order in which data from days 31-35 is released to the object store.

For example: If no data has been accessed or modified since creation, then the data from interval 0 will be released and the data from intervals 1-6 will remain on the SSDs. If, on the other hand, 8 TB of data is written every day, meaning that 40 TB of data is written in each interval (as shown below), then the last two intervals, i.e., data written, accessed, or modified in a total of 10 days will be kept on the SSD, while other data will be released to the object-store.

Interval	Interval 0	Interval 1	Interval 2	Interval 3	Interval 4	Interval 5	Interval 6
Date span	Jan 1-5	Jan 6-10	Jan 11-15	Jan 16-20	Jan 21-25	Jan 26-30	Jan 31-Feb 4
Capacity	40 TB	40 TB	40 TB	40 TB	40 TB	40 TB	40 TB
Overall SSD capacity assigned to the time span between this interval and now	280	240	200	160	120	80	40

Now consider the following filesystem scenario, where the whole SSD storage capacity of 100 TB is utilized in the first 3 intervals:

Interval	Interval 0	Interval 1	Interval 2	Interval 3	Interval 4	Interval 5	Interval 6
Date span	Jan 1-5	Jan 6-10	Jan 11-15	Jan 16-20	Jan 21-25	Jan 26-30	Jan 31-Feb 4
Capacity	60 TB	30 TB	10 TB				

When much more data is written and there is insufficient SSD capacity for storage, the data from interval 0 will be released when the 100 TB capacity is reached. This represents a violation of the Retention Period. In such a situation, it is also possible to either increase the SSD capacity or reduce the Retention Period.

Tiering cue

The tiering process (the tiering of data from the SSDs to the object stores) is based on when data is created or modified. It is managed similar to the Retention Period, with the data timestamped in intervals. The length of each interval is the size of the user-defined Tiering Cue. The Content Software for File system maintains 3 such intervals at any given time, and always tiers the data in the third interval.

Note: While the data release process is based on timestamps of access, creation, or modification, the tiering process is based only on the timestamps of the creation or modification.

Note: These timestamps are per 1 MB chunk and not the file timestamp.

For example: If the Tiering Cue is 1 day, then the data will be classified according to the following timeline for a system that starts working on January 1:

Interval 1	Interval 2	Interval 3
January 1, 00:00-24:00	January 2, 00:00-24:00	January 3, 00:00-24:00
DATA TIERING		

Since the tiering process applies to data in the first interval in this example, the data written or modified on January 1 will be tiered to the object store on January 3. Consequently, data will never be tiered before it is at least 1 day old (which is the user-defined Tiering Cue), with the worst case being the tiering of data written at the end of January 1 at the beginning of January 3.

Note: The Tiering Cue default is 10 seconds and cannot exceed 1/3 of the Data Retention period.

Breaks in retention period or tiering cue policies

If it is not possible to maintain the defined Retention Period or Tiering Cue policies, a *TieredFilesystemBreakingPolicy* event will occur, and old data will be released in order to free space on the SSDs. Users are alerted to such a situation through an *ObjectStoragePossibleBottleneck* event, enabling them to consider either raising the bandwidth or upgrading the object store performance.

Object-store direct mount option

Regardless of the time-based policies, it is possible to use a special mount option `obs_direct` to bypass the time-based policies. Any creation/writing of files from a mount point with this option will mark it to release as soon as possible, before taking into account other files retention policies. The data extents of the files are still first written to the SSD but get precedence on releasing to the object store.

In addition, any read done through such a mount point will read the extents from the object-store and will not be kept persistently on the SSD (it still goes through the SSD, but is released immediately before any other interval).

Note: In AWS, this mode should only be used for importing data. It should not be used for general access to the filesystem as any data read via this mount point would be immediately released from the SSD tier again. This can lead to excessive S3 charges.

Data management in tiered filesystems

The system behavior when tiering, accessing, or deleting data in tiered filesystems is described.

Overview

In tiered filesystems, the hot data resides in SSDs and warm data resides in object stores. When tiering, the Content Software for File system is highly efficient in terms of:

- Tiering only the subset of a file which is not accessed frequently, and not keeping infrequently-accessed portions of a file on SSDs.
- Gathering several subsets of different files and tiering them together to the object store (usually 64 MB objects), thereby providing a huge performance boost when working with object stores.
- When accessing data that resides just on the object store, only the required portion is retrieved from the object-store, regardless of the entire object it was tiered as part of it.

When data is logically freed, it can be reclaimed. Data is logically freed when it has been modified or deleted and is not being used by any snapshot.

Note: Only data which is not logically freed is taken into account for licensing purposes.

Space reclamation in tiered filesystems

Reclaming space in tiered filesystems has both an SSD and an object store methodology. For the details of both, see:

- SSD space reclamation (on page 74).
- Object store space reclamation (on page 74).

SSD space reclamation

For logically freed data which resides on SSD, the Content Software for File system immediately deletes the data from the SSD (leaving the physical space reclamation for the SSD erasure technique).

Object store space reclamation

For the object store, merely deleting the data from the object store is insufficient, since it might involve downloading up to 64 MB object and re-uploading most of the data just for a very small portion (even 4 KB) of the object.

To overcome this inefficiency, the Content Software for File system reclaims object store space in the background and will allow for 7%-13% more object store usage than required. In this way, for each filesystem that exceeds this 13% threshold, the 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). Moreover, the Content Software for File 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.

Note: If tuning of the system interaction with the object store is required (such as object size, reclamation threshold numbers, or the object store space reclamation is not fast enough for the workload), .contact customer support.

Note: Object store space reclamation is only relevant for object-store buckets used for tiering (defined as local) and not for buckets used for backup-only (defined as remote).

Object tagging

Note: To enable the object-tagging capability the Content Software for File object-store entity should be configured as such using **`enable-upload-tags`** parameter in either of the **`weka fs tier s3`** add/update CLI commands.

Whenever Content Software for File uploads objects to the object store, it classifies them using tags. It is useful to carry further lifecycle management rules via the object-store based on these tags (e.g., transfer objects of a specific filesystem to/from Glacier).

Tag	Description
`wekaBlobType`	The Weka-internal type representation of the object. One of: `DATA`, `METADATA`, `METAMETADATA`, `LOCATOR`, `RELOCATIONS`
`wekaFsId`	The filesystem ID (a combination of the filesystem ID and the cluster GUID uniquely identifies a filesystem).
`wekaGuid`	The cluster GUID
`wekaFsName`	The name of the filesystem that uploaded this object.

The object-store must support S3 object-tagging and might require additional permissions to use object tagging.

The following extra permissions are required when using AWS S3:

- `s3:PutObjectTagging`
- `s3:DeleteObjectTagging`

Note: Additional charges may apply when using AWS S3.

Transition between tiered and SSD-only filesystems

How to transition from an SSD-only to a tiered filesystem and the other way around.

Transition from SSD-only filesystem to tiered filesystem

An SSD-only filesystem can be reconfigured as a tiered filesystem by attaching an object store. In such a situation, the default is to maintain the filesystem size. In order to increase the filesystem size, the total capacity field can be modified, while the existing SSD capacity remains the same.

Note: Once an SSD-only filesystem has been reconfigured as a tiered filesystem, all existing data will be considered to belong to interval 0 and will be managed according to the 7-interval process. This means that the release process of the data created before the reconfiguration of the filesystem is performed in an arbitrary order and does not depend on the timestamps.

Transition from tiered filesystem to SSD-only filesystem

A tiered filesystem can be un-tiered (and only use SSDs) by detaching its object stores. This will copy the data back to the SSD.

Note: The SSD must have sufficient capacity, this means that the allocated SSD capacity should be at least the total capacity used by the filesystem.

For more information, refer to Attaching/Detaching Object Stores.

Chapter 6: Snapshots

Snapshots enable the saving of a filesystem state to a directory and can be used for backup, archiving and testing purposes.

About snapshots

Snapshots allow the saving of a filesystem state to a `.snapshots` directory located under the `root` filesystem. They can be used for:

- Physical backup: The snapshot directory can be copied into a different storage system, possibly on another site, using either the Content Software for File system Snap-To-Object feature or a third-party software.
- Logical backup: Periodic snapshots enable filesystem restoration to a previous state if logical data corruption occurs.
- Archive: Periodic snapshots enable the accessing of a previous filesystem state for compliance or other needs.
- DevOps environments: Writable snapshots enable the execution of software tests on copies of the data.

Snapshots have no impact on system performance and can be taken for each filesystem while applications are running. They consume minimal space, according to the actual differences between the filesystem and the snapshots, or between the snapshots, in 4K granularity.

By default, snapshots are read-only, and any attempt to change the content of a read-only snapshot returns an error message.

It is possible to create a writable snapshot or update an existing snapshot to be writable. However, a writable snapshot cannot be changed to a read-only snapshot.

The Content Software for File system supports the following snapshot operations:

- View snapshots.
- Create a snapshot of an existing filesystem.
- Delete a snapshot.
- Access a snapshot under a dedicated directory name.
- Restore a filesystem from a snapshot.
- Make snapshots writable.
- Create a snapshot of a snapshot (relevant for writable snapshots, or for read-only snapshots before being made writable).

- List of snapshots and obtaining their metadata.
- Schedule automaticsnapshots. See Snapshot management.

Note: The number of snapshots per system is limited to 4,096 (the live filesystem consumes one of the total snapshots count).

Managing snapshots using the GUI

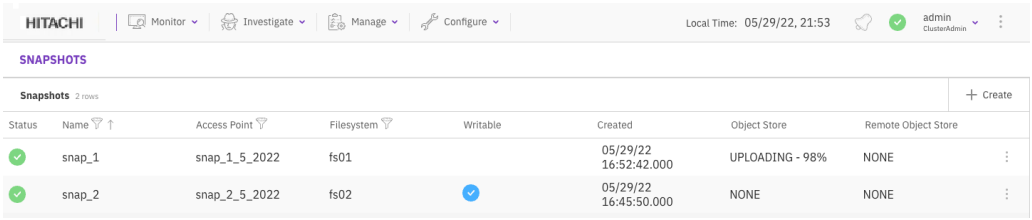
How to manage snapshots using the GUI.

Viewing snapshots using the GUI

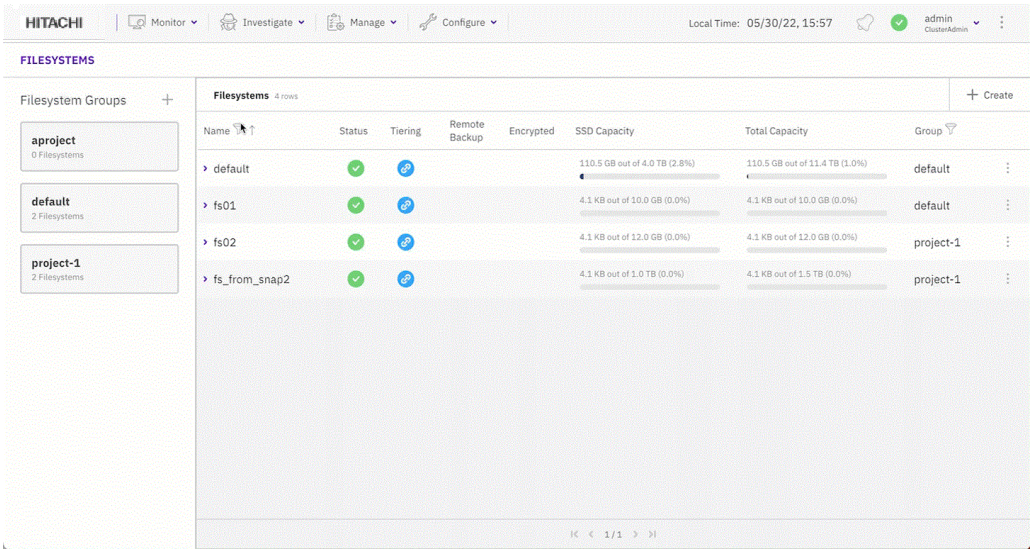
To view the snapshot of filesystem

Procedure

1. To display all snapshots, select **Manage > Snapshots** from the menu. The Snapshots page opens.



2. To display a snapshot of a selected filesystem, do one of the following:
   - Select the Filesystem filter. Then, select the filesystem from the list.
   - From the menu, select **Manage > Filesystems**. From the filesystem, select the three dots, and from the menu, select **Go To Snapshot**.



Creating a snapshot using the GUI

Before you begin

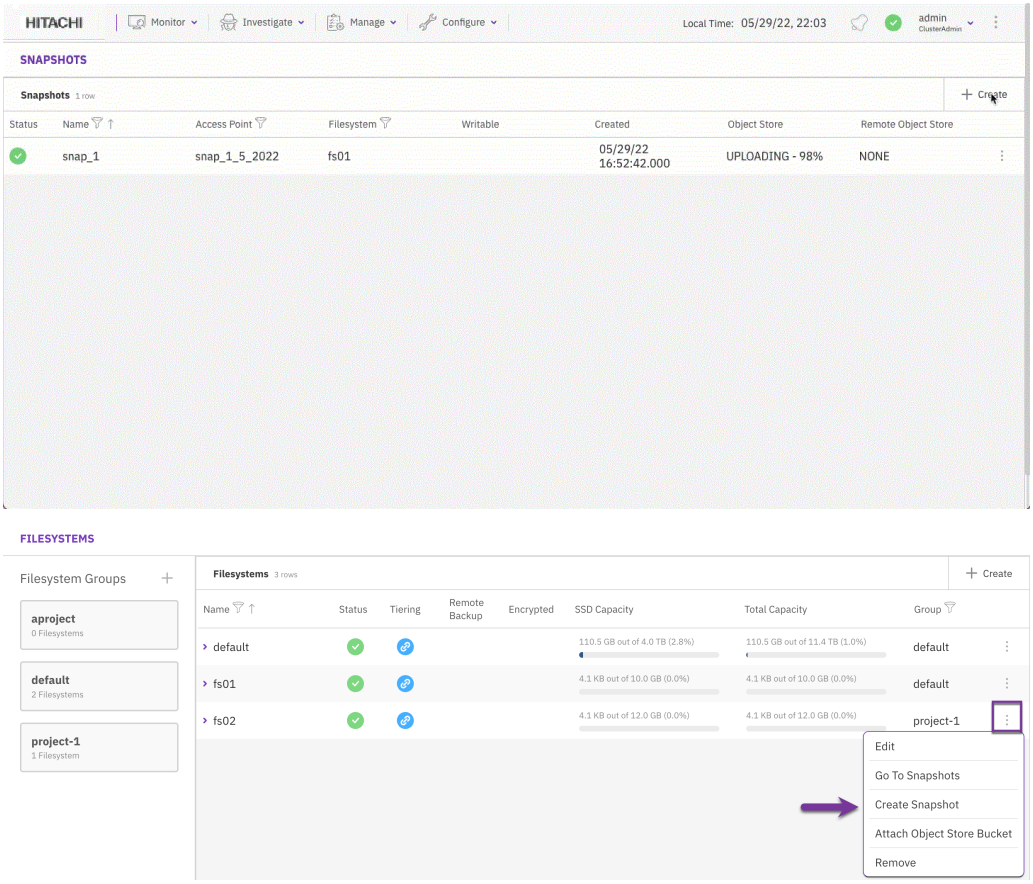
You can create a snapshot from the Snapshots page or directly from the Filesystems page.

Create a directory for filesystem-level snapshots that serves as the access point for snapshots.

Procedure to create a snapshot.

Procedure

1. Do one of the following:
   - From the menu, select **Manage > Snapshots**. From the Snapshots page, select +Create. The Create Snapshot dialog opens.
   - From the menu, select **Manage > Filesystems**. From the Filesystems page, select the three dots, and from the menu, select **Create Snapshot**. (The source filesystem is automatically set.)



2. On the Create Snapshot dialog set the following properties:
   - **Name**: A unique name for the filesystem snapshot.
   - **Access Point**: A name of the newly-created directory for filesystem-level snapshots that serves as the snapshot's access point.
   - **Writable**: Determines whether to set the snapshot to be writable.
   - **Source Filesystem**: The source filesystem from which to create the snapshot.
   - **Upload to local object store**: Determines whether to upload the snapshot to a local object store. You can also upload the snapshot later (see Snap-To-Object).
   - **Upload to remote object store**: Determines whether to upload the snapshot to a remote object store. You can also upload the snapshot later.
3. Select Create.

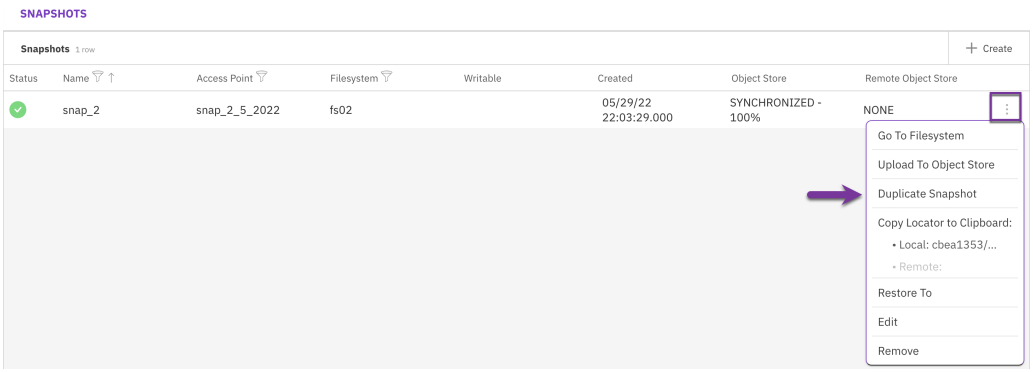
Duplicate a snapshot

You can duplicate a snapshot (clone), which enables creating a writable snapshot from a read-only snapshot.

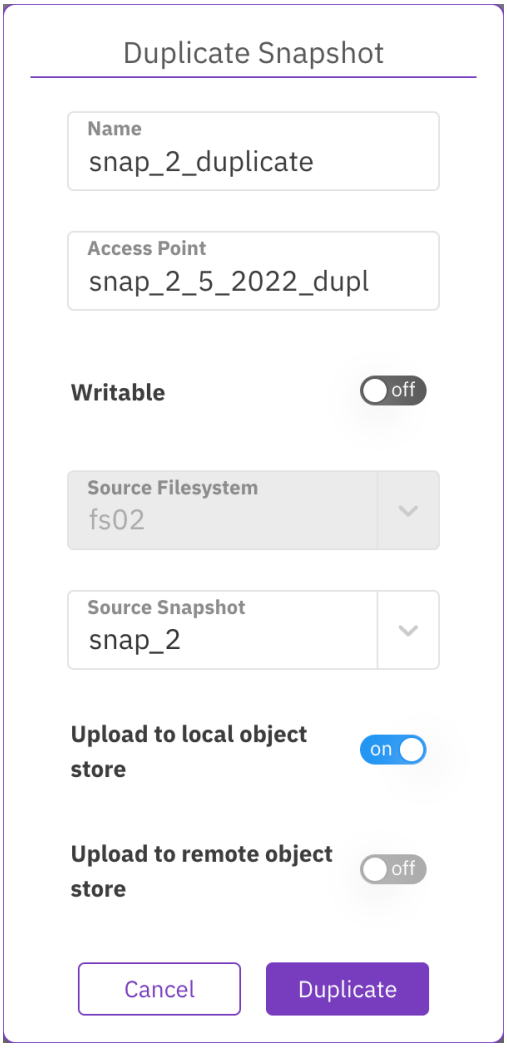
Procedure

1. From the menu, select **Manage > Snapshots**.

2. From the Snapshots page, select the three dots of the snapshot you want to duplicate, and from the menu, select **Duplicate Snapshot**.



3. In the Duplicate Snapshot dialog, set the properties the same way you create a snapshot. The source filesystem and source snapshot are already set.
4. Select **Duplicate**.



Deleting a snapshot using the GUI

Before you begin

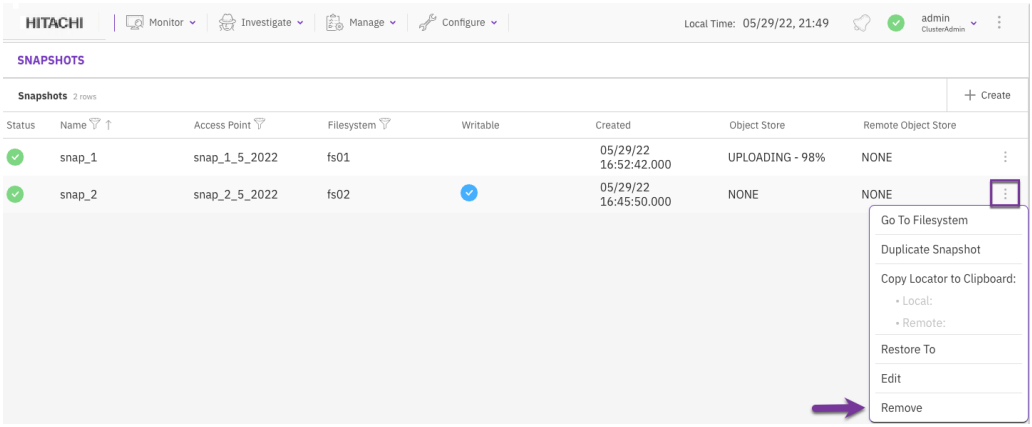
When deleting a snapshot, consider the following guidelines:

- Deleting a snapshot in parallel to a snapshot upload to the same filesystem is not possible. Uploading a snapshot to a remote object store can take time. Therefore, it is advisable to delete the desired snapshot before uploading it to the remote object store.
- When uploading snapshots to both local and remote object stores. While the local and remote uploads can progress in parallel, consider the case of a remote upload in progress, then a snapshot is deleted, and later a snapshot is uploaded to the local object store. In this scenario, the local snapshot upload waits for the pending deletion of the snapshot (which happens only once the remote snapshot upload is done).

Procedure to delete a snapshot.

Procedure

1. From the menu, select **Manage > Snapshots**.
2. From the Snapshots page, select the three dots of the snapshot you want to delete, and from the menu, select **Remove**.
3. In the Deletion Of Snapshot message, select **Yes** to delete the snapshot.



Restore a snapshot to a filesystem or another snapshot

Before you begin

Restoring a snapshot to a filesystem or another snapshot (target) modifies the data and metadata of the target.

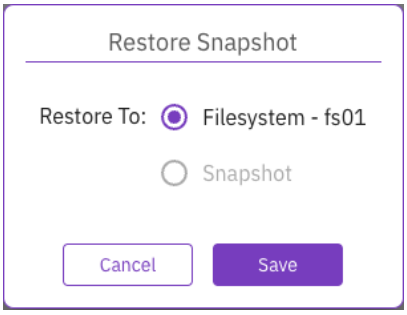
If you restore the snapshot to a filesystem, make sure to stop the IO services of the filesystem during the restore operation.

Procedure to restore a snapshot.

Procedure

1. From the menu, select **Manage > Snapshots**.

2. From the Snapshots page, select the three dots of the snapshot you want to restore, and from the menu, select **Restore To**.
3. In the Restore To dialog, select the destination: **Filesystem or Snapshot**.
4. Select **Save**.

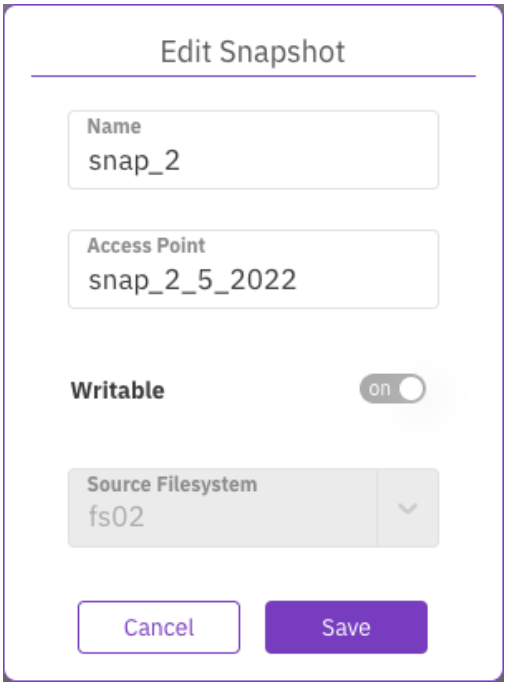


Updating a snapshot using the GUI

You can update the snapshot name and access point properties.

Procedure

1. From the menu, select **Manage > Snapshots**.
2. From the Snapshots page, select the three dots of the snapshot you want to update, and from the menu, select **Edit**.
3. Modify the **Name** and **Access Point** properties as required.
4. Select **Save**.



Chapter 7: Working with snapshots

How to work with snapshots using the GUI.

Snapshot management

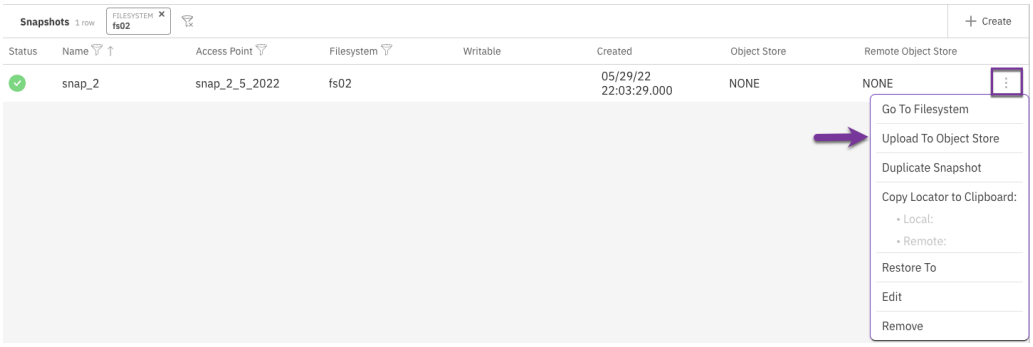
For information on snapshot viewing, creation, updating, deletion and restoring a filesystem from a snapshot, refer to Managing snapshots (on page 78).

Uploading a snapshot using the GUI

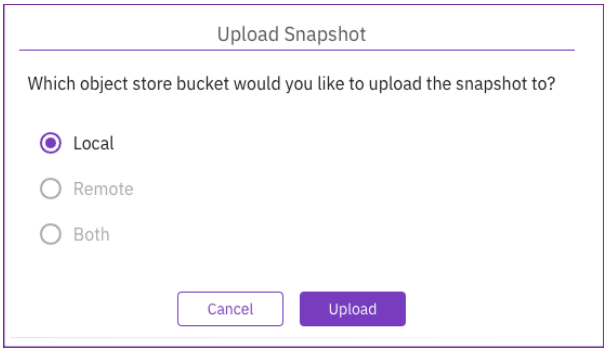
You can upload a snapshot to a local, remote, or both object store buckets.

Procedure

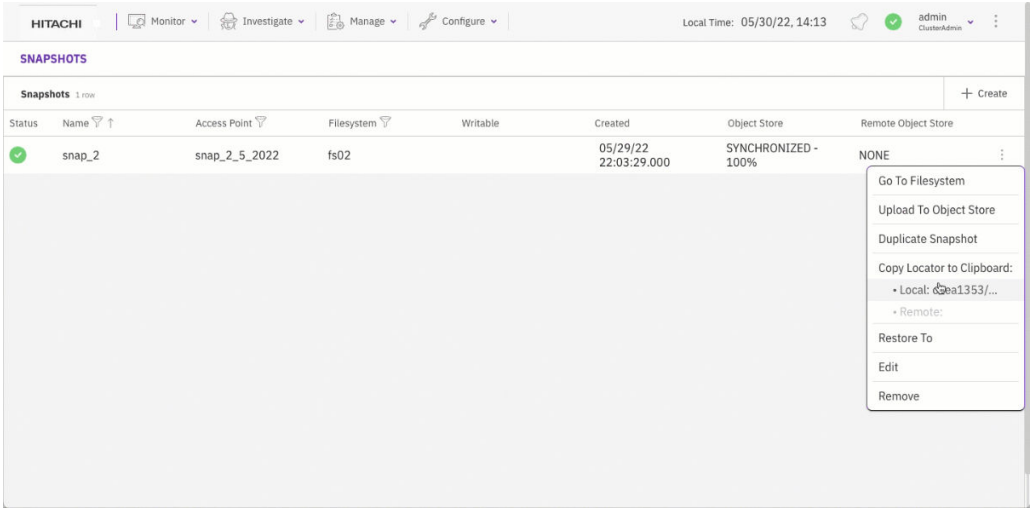
1. From the menu, select **Manage > Snapshots**.
2. Select the three dots on the right of the required snapshot. From the menu, select **Upload To Object Store**.



3. In the Upload Snapshot dialog, select the target object store bucket: Local, Remote, or Both.



4. Select **Upload**.
5. In the confirmation message, select **Yes**. The snapshot is uploaded to the target object store bucket.
6. To copy the snapshot locator, select the three dots on the right of the required snapshot. From the menu, select **Copy Locator to Clipboard**. Then, save the locator in a dedicated file.

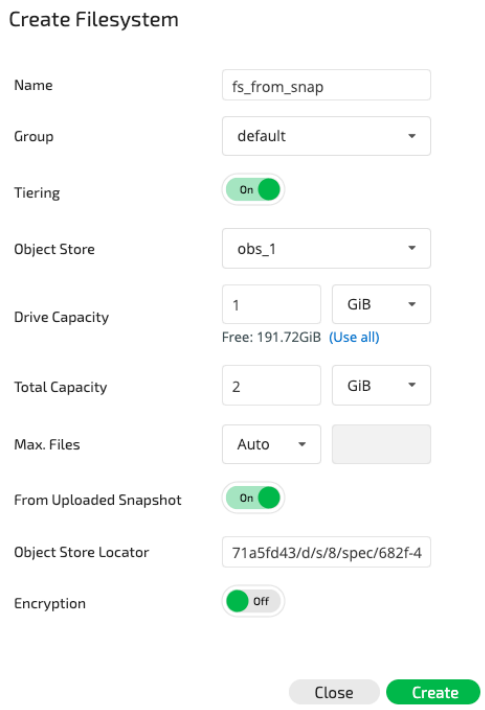


Creating a filesystem from an uploaded snapshot

To create a filesystem from an uploaded snapshot:

Procedure

1. Switch the **From Uploaded Snapshot** field in the **Filesystem Creation** dialog box to **On**. The **Create Filesystem** dialog box is displayed.



2. Define all the fields and enter the location of the snapshot to be used in the **Object Store Locator** field.

Deleting snapshots residing on an object store

Deleting a snapshot from a filesystem that uploaded it will remove all of its data from the object store.

Caution: If the snapshot has been (or is) downloaded and used by a different filesystem, that filesystem will stop functioning correctly, data might be unavailable, and errors might occur when accessing the data.

It is possible to either un-tier or migrate the filesystem to a different object store bucket before deleting the snapshot it has downloaded.

Chapter 8: Snap-to-object

This page describes the Snap-To-Object feature, which enables the movement of all the data of a specific snapshot to an object store.

The Snap-To-Object feature enables the committing of all the data of a specific snapshot, including file system metadata, every file, and all associated data to an object store. You can use the full snapshot data to restore the data on the Content Software for File cluster or another cluster.

Snap-To-Object feature use cases

The Snap-To-Object feature is helpful for a range of use cases, as follows:

- On-premises and cloud use cases
  - External backup of data (on page 87)
  - Archiving data (on page 88)
  - Asynchronous data replication (on page 88)
- Cloud-only use cases
  - Cloud pause/restart (on page 88)
  - Data protection against cloud availability zone failures (on page 89)
  - Migration of filesystems to another region (on page 89)

External backup of data

Suppose it is required to recover data stored on a Content Software for File filesystem due to a complete or partial loss of the data within it. You can use a data snapshot saved to an object store to recreate the same data in the snapshot on the same or another Content Software for File cluster.

This use case supports backup in any of the following Content Software for File system deployment modes:

Local Object Store

The Content Software for File cluster and object store are close to each other and will be highly performant during data recovery operations. The Content Software for File cluster can recover a filesystem from any snapshot on the object store for which it has a reference locator.

Remote Object Store

The Content Software for File cluster and object store are located in different geographic locations, typically with longer latencies between them. In such a deployment, you can send snapshots to both local and remote object stores.

Note: This deployment type requires supporting the latency of hundreds of milliseconds. For performance issues on Snap-To-Object tiering cross-interactions/resonance, contact customer support.

Local Object Store Replicating to a Remote Object Store

A local object store in one datacenter replicates data to another object store using the object store system features, such as AWS S3 cross-region replication.

This deployment provides both integrated tiering and Snap-To-Object local high performance between the Weka object store and the additional object store. The object store manages the data replication, enabling data survival in multiple regions.

Note: This deployment requires ensuring that the object store system perfectly replicates all objects on time to ensure consistency across regions.

Archiving data

The periodic creation of snapshots and uploading of the snapshots to an object store generates an archive, allowing the accessing of past copies of data.

When any compliance or application requirement occurs, it is possible to make the relevant snapshot available on a Content Software for File cluster and view the content of past versions of data.

Asynchronous data replication

Combining a local cluster with a replicated object store in another data center allows for the following use cases:

- Disaster recovery: where you can take the replicated data and make it available to applications in the destination location.
- Backup: where you can take multiple snapshots and create point-in-time images of the data that can be mounted and specific files may be restored.

Cloud pause/restart

In a public cloud, with a Content Software for File cluster running on compute instances with local SSDs, sometimes the data needs to be retained, even though ongoing access to the Content Software for File cluster is unnecessary. In such cases, using Snap-To-Object can save the costs of compute instances running the Content Software for File system.

To pause a cluster, you need to take a snapshot of the data and then use Snap-To-Object to upload the snapshot to an S3 compliant object store. When the upload process is complete, the Content Software for File cluster instances can be stopped, and the data is safe on the object store.

To re-enable access to the data, you need to form a new cluster or use an existing one and download the snapshot from the object store.

Data protection against cloud availability zone failures

This use case ensures data protection against cloud availability zone failures in the various clouds: AWS Availability Zones, Google Cloud Platform (GCP) Zones, and Oracle Cloud Infrastructure (OCI) Availability Domains.

In AWS, for example, the Content Software for File cluster can run on a single availability zone, providing the best performance and no cross-AZ bandwidth charges. Using Snap-To-Object, you can take and upload snapshots of the cluster to S3 (which is a cross-AZ service). In this way, if an AZ failure occurs, a new Content Software for File cluster can be created on another AZ, and the last snapshot uploaded to S3 can be downloaded to this new cluster.

Migration of filesystems to another region

Using Content Software for File snapshots uploaded to S3 combined with S3 cross-region replication enables the migration of a filesystem from one region to another.

Chapter 9: Snap-to-object in data lifecycle management

Snap-To-Object and data lifecycle management both use SSDs and object stores for the storage of data. In order to save both storage and performance resources, the Content Software for File system uses the same paradigm for holding SSD and object store data for both lifecycle management and Snap-To-Object. This can be implemented for each filesystem using one of the following schemes:

1. Data resides on the SSDs only and the object store is used only for the various Snap-To-Object use cases, such as backup, archiving and *bursting*. Bursting is an application relationship between a private and public cloud in which the application operating in the private cloud bursts to the public cloud when necessary to meet peak demands. In this case, for each filesystem, the allocated SSD capacity should be identical to the filesystem size and the data Retention Period should be defined as the longest time possible, specifically, 5 years. The Tiering Cue should be defined using the same considerations as in data lifecycle management, based on IO patterns. In this scheme, the applications work all the time with a high-performance SSD storage system and use the object store only as a backup device.
2. Use of Snap-To-Object on filesystems with active data lifecycle management between the object store and the SSDs. In this case, objects in the object store will be used for both tiering of all data and for backing-up the data using Snap-To-Object, specifically, whenever possible, the Content Software for File system will use the same object for both purposes, thereby eliminating the need to acquire additional storage and to unnecessarily copy data.

Note: When using Snap-To-Object to rehydrate data from an object store, some of the metadata may still be in the object store until it is accessed for the first time.

Chapter 10: Quota management

This page describes how to manage quotas to alert or restrict usage of the WekaFS filesystem.

Overview

There are several levels on the Content Software for File system where capacity usage can be restricted.

1. On an organization level: Set a different organization to manage its own filesystems, where quotas for an organization can be set, as described in the organization's usage and quota management section.
2. On a filesystem level: Set a different filesystem per department/project.
3. On a directory level: Set a different quota per project directory (useful when users are part of several projects) or per user home directory.

Directory quotas

The organization admin can set a quota on a directory. Setting a quota will start the process of counting the current directory usage. Until this process is done, the quota will not be taken into account (for empty directories, this process is instantly done).

Note: Currently, a mount point to the relevant filesystem is required to set a quota on a directory, and the quota set command should not be interrupted until the quota accounting is over.

The organization admin sets quotas to inform/restrict users from using too much of the filesystem capacity. For that, only data in the user's control is taken into account. Hence, the quota doesn't count the overhead of the protection bits and snapshots. It does take into account data&metadata of files in the directory, regardless if tiered or not.

Working with quotas

When working with quotas, consider the following:

- Currently, to set a quota, the relevant filesystem must be mounted on the host where the set quota command is to be run.
- When setting a quota, you should go through a new mount-point. Meaning, if you are using a host that has mounts from Weka versions before 3.10, first unmount all relevant mount point and then mount them again.
- Quotas can be set within nested directories (up to 4 levels of nested quotas are supported) and over-provisioned under the same directory quota tree. E.g., /home can have a quota of 1TiB, and each user directory under it can have a quota of 10GiB, while there are 200 users.
- Before a directory is being deleted, its quota must be removed. A directory tree cannot be deleted without removing all the inner directories quotas beforehand. Note, default (parent) quotas are set as quotas at the directory creation and the actual quota needs to be removed before the directory is deleted (not the default quota of the parent directory)
- Moving files (or directories) between two directories with quotas, into a directory with a quota, or outside of a directory with a quota is not supported. The WekaFS filesystem returns EXDEV in such a case, which is usually converted by the operating system to copy&delete but is OS-dependent.
- Quotas and hardlinks:
  - An existing hardlink is not counted as part of the quota.
  - Once a directory has a quota, it is not allowed to create a hardlink to files residing under directories with different (or without) directory quotas.
- Restoring a filesystem from a snapshot turns the quotas back to the configuration at the time of the snapshot.
- Creating a new filesystem from a snap-2-obj does not preserve the original quotas.
- When working with enforcing quotas along with a `writecache` mount-mode, similarly to other POSIX solutions, getting above the quota might not sync all the cache writes to the backend servers. Use `sync`, `syncfs`, or `fsync` to commit the cached changes to the system (or fail due to exceeding the quota).

Integration with the df utility

When a hard quota is set on a directory, running the `df` utility will consider the hard quota as the total capacity of the directory and provide the use% relative to the quota. This can help users understand their usage and how close they are to the hard quota.

Note: The `df` utility behavior with quotas is currently global to the Content Software for File system. To change the global behavior, contact customer support.

Chapter 11: NFS

How the Content Software for File system enables file access through the NFS protocol instead of through the client. The Content Software for File system supports NFS v3.

Workflow: Deploy NFS service with a Content Software for File client software

The Content Software for File system supports the NFSv3, NFSv4.0, and NFSv4.1 protocols. The NFS protocol allows client hosts to access the Content Software for File filesystem without installing Content Software for File's client software using the standard NFS implementation of the client host operating system. While this implementation is easier to deploy, it does not compare in performance to the Content Software for File client.

In order to implement NFS service from a Content Software for File cluster, the following steps must be implemented:

Step	Method of Implementation
Define a set of hosts that will provide the NFS service, which can be the whole cluster or a subset of the cluster.	Define an interface group.
Define Ethernet ports on each of the defined hosts that will be used to provide the NFS service.	Define an interface group.
Allocate a pool of IP addresses that will be used by the Content Software for File software to provide the NFS service.	Define an interface group.
Define a Round-robin DNS name that resolves to the floating IPs.	On the local DNS service configuration; does not involve Content Software for File management.
Define the list of client hosts that have access permissions to the NFS filesystems.	Create a client permission group.
Configure the client hosts and the filesystems that they can access.	Create a client permission group.

Step	Method of Implementation
Mount the file systems on the client hosts using the NFS mount operating system support.	On the client operating system; does not involve Content Software for File management. Content Software for File management.

Defining the NFS networking configuration (interface groups)

You can add only a single port to an interface group. To support High Availability (HA) in NFS, create two interface groups. On each interface group, assign the host ports.

To ensure that a single point of failure is not created in the switch, consider the network topology (switches) when assigning the other host ports to these interface groups.

Implementing NFS service from a Content Software for File cluster

In order to define the NFS service, one or more interface groups must be defined. An interface group consists of the following:

- A collection of Content Software for File hosts with an Ethernet port for each host, where all the ports must belong to the same layer 2 subnets.
- A collection of floating IPs that serve the NFS protocol on the hosts and ports. All IP addresses must belong to the layer 2 subnet above.
- A routing configuration for the IPs which must comply with the IP network configuration.

Up to 10 different Interface groups can be defined, where multiple interface groups can be used if the cluster needs to connect to multiple layer 2 subnets. Up to 50 hosts can be defined in each interface group.

The Content Software for File system will automatically distribute the IP addresses evenly on each host and port. On failure of the host, the Content Software for File system will reasonably redistribute the IP addresses associated with the failed host on other hosts. To minimize the effect of any host failures, it is recommended to define sufficient floating IPs so that the system can assign four floating IPs per host.

Note: The Content Software for File system will configure the host IP networking for the NFS service on the host operating system. It should not be defined by the user.

Configuring the round-robin DNS server

To ensure that the various NFS clients will balance the load on the various Content Software for File hosts serving NFS, it is recommended to define a Round-robin DNS entry which will resolve to the list of floating IPs, ensuring that client loads will be equally distributed across all hosts.

Note: Make sure to set the Time to Live (TTL) for all A records assigned to the NFS servers to 0 (zero), this ensures that the IP won't be cached by the client or the DNS server.

Defining NFS access control (client access groups)

In order to control which host can access which file system, NFS client permission groups must be defined. Each NFS client permission group contains:

- A list of filters for IP addresses or DNS names of clients that can be connected to the Content Software for File system using NFS.
- A collection of rules that control access to specific filesystems.

Configuring NFS on the client

The NFS mount should be configured on the client host using the standard NFS stack operating system. The NFS server IP address should point to the Round-Robin DNS name defined above.

NFS service load balancing and resiliency

The Content Software for File NFS service is a scalable, fully load-balanced, and resilient service that provides continuous service through failures of any kind.

Scalability is implemented by defining many hosts that serve the NFS protocol, thereby enabling the scaling of performance by adding more hosts to the interface group.

Load balancing is implemented using floating IPs. By default, the floating IPs are evenly distributed over all the interface group hosts/ports. When different clients resolve the DNS name into an IP service, each of them receives a different IP address, thereby ensuring that different clients will access different hosts. This allows the Content Software for File system to scale and service thousands of clients.

The same mechanism ensures the resiliency of the service. On a host failure, all IP addresses associated with the failed host will be reassigned to other hosts (using the Gratuitous Address Resolution Protocol (GARP) network messages) and the clients will reconnect to the new hosts without any reconfiguration or service interruption.

Managing NFS networking configuration (interface groups)

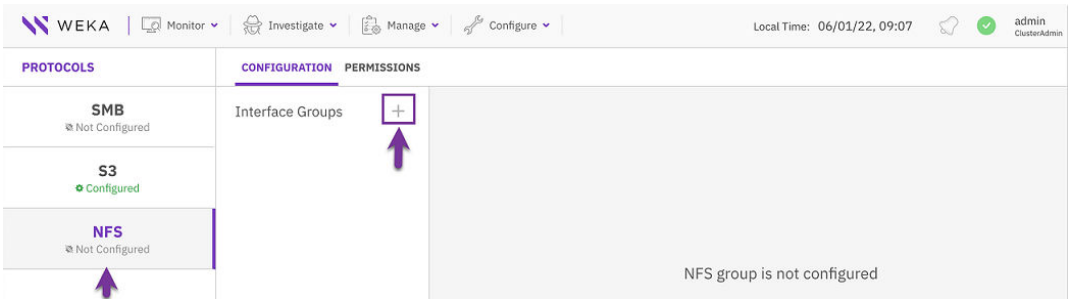
How to manage the NFS networking configuration (interface groups) using the GUI.

Creating interface groups using the GUI

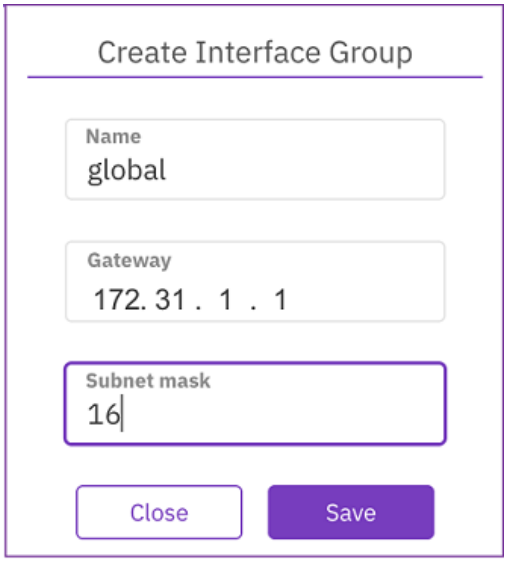
Interface Groups define the hosts and ports that provide the NFS service.

Procedure

1. From the menu, select **Manage > Protocols**.
2. On the left pane, select **NFS**.
3. In the Configuration tab, select the **+** sign near the Interface Groups title.



4. In the Create Interface Group dialog set the following properties:
   - **Name**: A unique interface group name (maximum 11 characters).
   - **Gateway**: A valid IP address of the gateway.
   - **Subnet mask**: The subnet mask in CIDR (Classless Inter-Domain Routing) format. For example, a value of 16 equals 255.255.0.0.
5. Select **Save**.



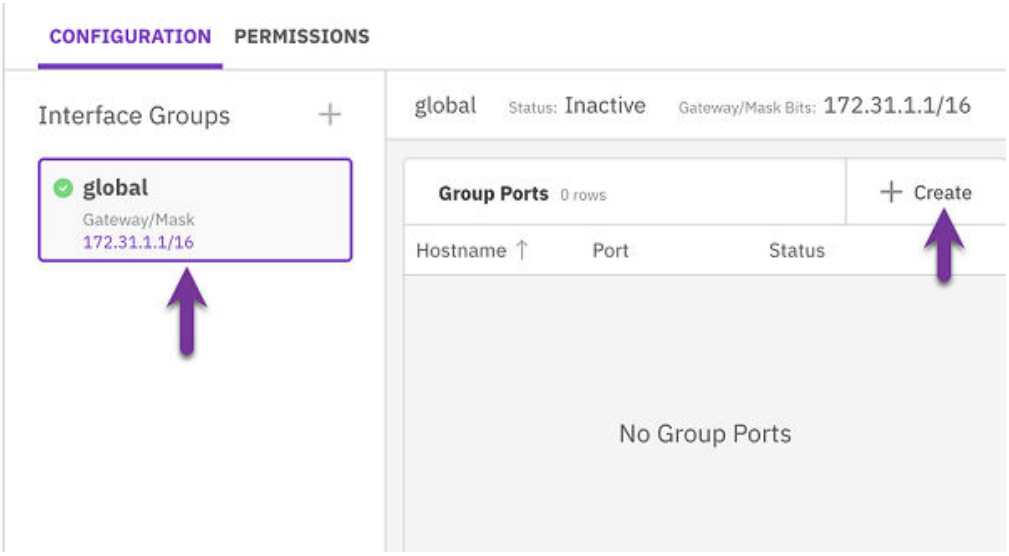
Setting interface group ports using the GUI

Once you create an interface group, set its ports.

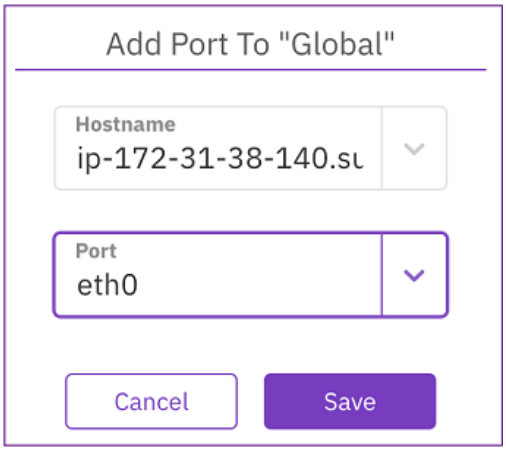
Procedure

1. In the Configuration tab, select the interface group.

2. In the Group Ports table, select **+Create**.



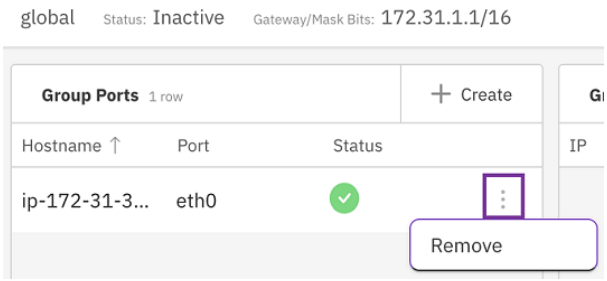
3. In the Add Port dialog, set the following properties:
   - **Hostname**: Select the host ID on which the port resides.
   - **Port**: Select the port from the list.



Removing an Interface Group Port using the GUI

Procedure

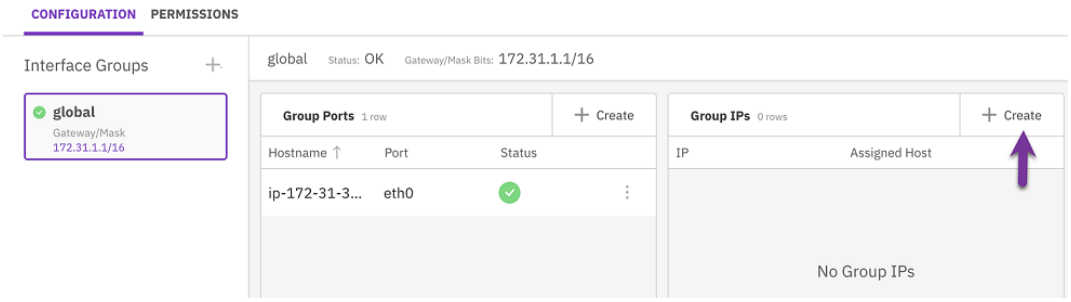
1. In the Configuration tab, select the interface group.
2. In the Group Ports table, select the three dots, and from the menu select **Remove**.



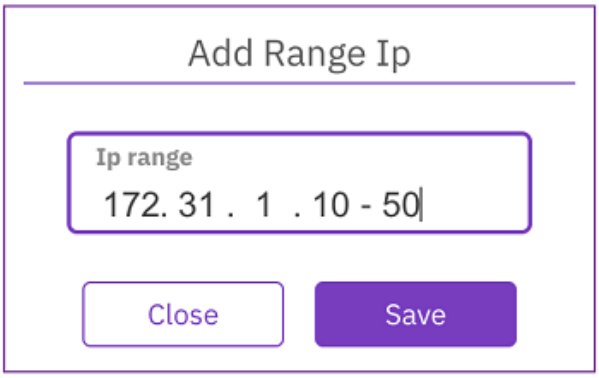
Setting interface group IPs using the GUI

Procedure

1. In the Configuration tab, select the interface group.
2. In the Group IPs table, select **+Create**.



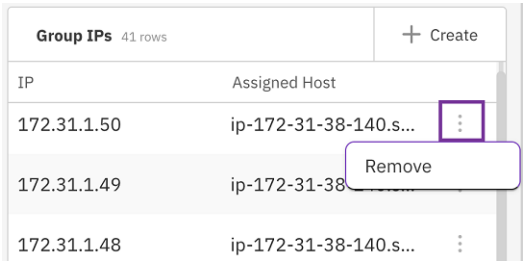
3. In the Add Rang IP dialog, set the relevant IP range.
4. Select **Save**.



Removing an Interface Group IPs using the GUI

Procedure

1. In the Configuration tab, select the interface group.
2. In the Group IPs table, select the three dots, and from the menu select **Remove**.



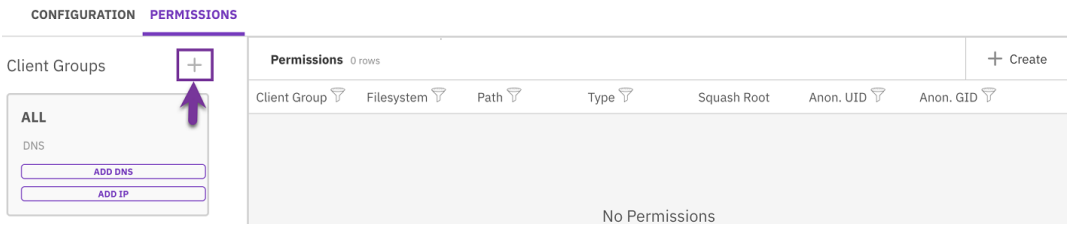
Managing NFS access control (client access groups)

How to manage NFS access control (client access groups) using the GUI.

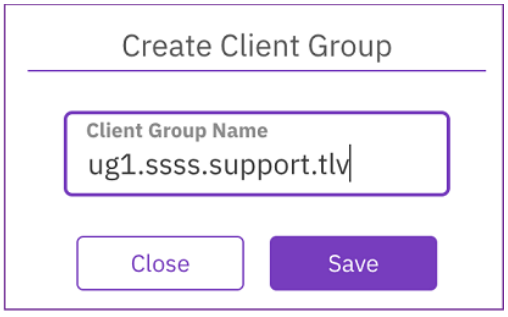
Defining client access groups using the GUI

Procedure

1. In the **Permissions** tab, select the **+** sign near the Client Groups title.



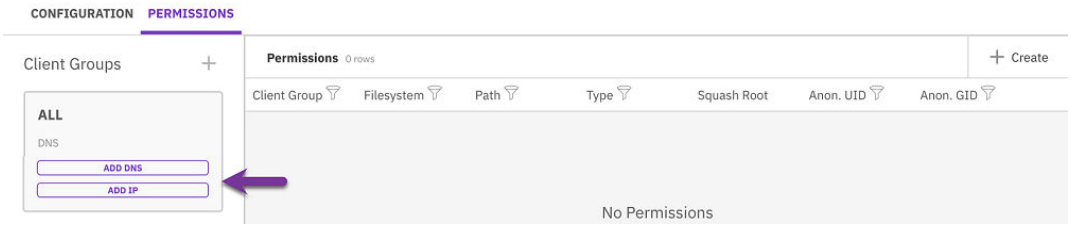
2. In the **Create Client Group** dialog, set the client group name (DNS server name).
3. Select **Save**.



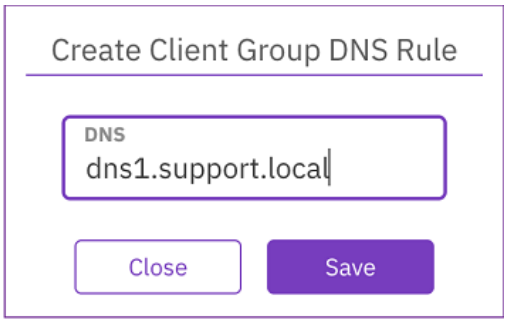
Managing client access groups using the GUI

Procedure

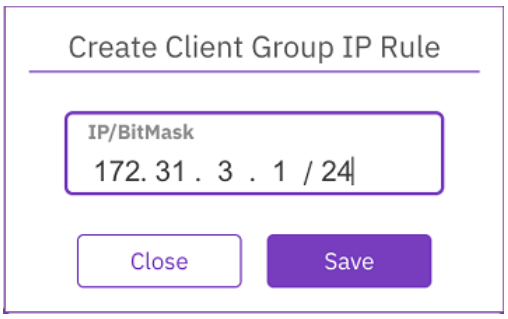
1. In the Permissions tab, select **ADD DNS** for the relevant Client Group.



2. In the **Create Client Group DNS Rule** dialog, set the DNS server name. Then, select **Save**.



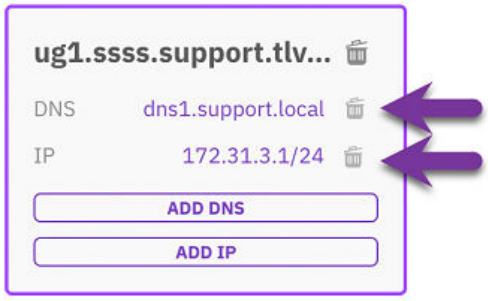
3. In the **Permissions** tab, select **ADD IP** for the relevant Client Group.
4. In the **Create Client Group IP Rule** dialog, set the IP address and bitmask. Then, select **Save**.



Removing the DNS or IP of a client group using the GUI

Procedure

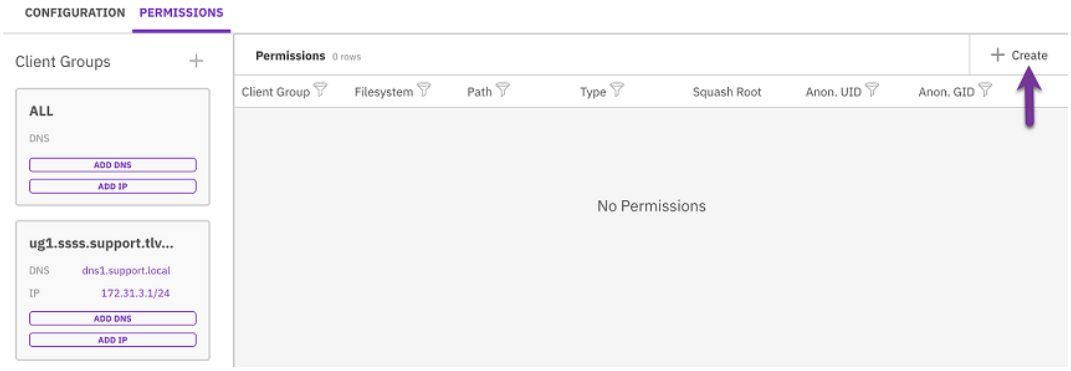
1. In the **Permissions** tab, select the **trash** symbol displayed next to the DNS or IP for the relevant Client Group.



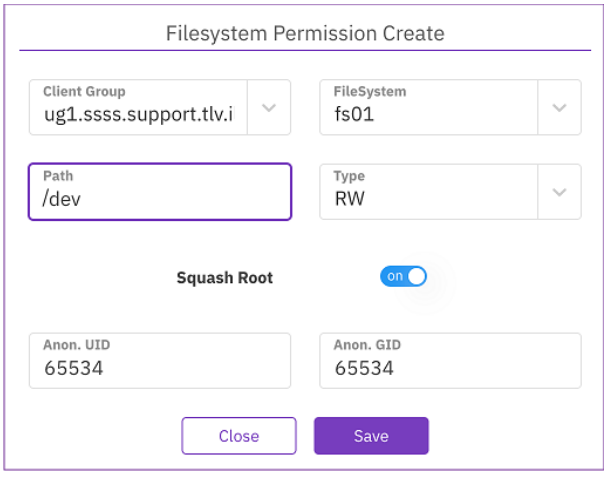
Managing NFS client permissions

Procedure

1. In the **Permissions** table, select **+Create**.



2. In the **Filesystem Permission Creation** dialog, set the following properties:
   - **Client Group**: The client group to which the permissions are applied.
   - **Filesystem**: The filesystem to which the permissions are applied. A filesystem set with required authentication cannot be used for NFS export.
   - **Path**: The exported directory path (root share).
   - **Type**: The access type: RO (read-only) or RW (read/write).
   - **Squash Root**: The squash mode that the system enforces with the client permission.
   - **Anon. UID**: Anonymous user ID. Only relevant for Root and All user squashing.
   - **Anon. GID**: Anonymous group ID. Only relevant for Root and All user squashing.
3. Select **Save**.



Chapter 12: SMB

The Content Software for File implementation of the SMB protocol for shared Windows clients is described.

About SMB

SMB (Server Message Block) is a network file sharing protocol that allows remote systems to connect to shared file and print services. Content Software for File's implementation is based on the open-source Samba package and provides support for SMB versions 2 and 3.

The Content Software for File implementation of SMB makes storage services available to Windows and macOS clients. Content Software for File provides shared access from multiple clients, including multi-protocol access to the same files from SMB, NFS, and Content Software for File native filesystem drivers.

SMB implementation key features

Implementation of the SMB feature in the Content Software for File system is scalable, resilient, and distributed.

- Scalable: The Content Software for File system currently supports an SMB cluster of between 3 to 8 hosts. These hosts run the SMB gateway service, while the backend filesystem can be any Content Software for File filesystem. Therefore, it is practically unlimited in size and performance.
- Resilient: The Content Software for File system implementation of SMB provides clustered access to files in a Content Software for File file store, enabling multiple servers to work together. Consequently, if a server failure occurs, another server is available to take over operations, thereby ensuring failover support and high availability. Content Software for File standard resiliency against failures also protects the SMB filesystems.
- Distributed: A Content Software for File implementation is distributed over a cluster, where all nodes in the cluster handle all SMB filesystems concurrently. Therefore, performance supported by SMB can scale with more hardware resources, and high availability is ensured.

SMB user-mapping

The Content Software for File ssystem SMB supports authentication by a single Active Directory with multiple trusted domains. The POSIX users (uid) and groups (gid) mapping for the SMB access must be resolved by the Active Directory.

The Content Software for File system pulls users and groups information from the Active Directory automatically and supports two types of id-mapping from the Active Directory:

- `RFC2307` where `uidNumber` and `gidNumber` must be defined in the AD user attributes.
- `rid` which creates local mapping with the AD users and groups.

Using `rid` mapping can ease the configuration, where user IDs are tracked automatically. All domain user accounts and groups are automatically available on the domain member, and no attributes need to be set for domain users and groups. On the other hand, if the `rid` AD range configuration changes, user mapping might change and result in wrong uids/gids resolution.

Active Directory attributes

The following are the Active Directory attributes relevant for users according to `RFC2307`:

AD Attribute	Description
uidNumber	0-4290000000
gidNumber	0-4290000000; must correlate with a real group.

The following are the Active Directory attributes relevant for groups of users according to `RFC2307`:

AD Attribute	Description
gidNumber	0-4290000000

The range specified above is the default configuration for the Content Software for File system for the AD server IDs and can be changed. This is the main AD range (if additional trusted domains are defined).

To avoid ID overlapping and collisions, set the range or ranges (for multiple domains).

When joining multiple domains, it is required to set the ID range for each of them, and the ranges cannot overlap. There is also a (configurable) default mapping range for users not part of any domain.

For more information, see Active Directory attributes.

Configuring SMB

Refer to the CLI commands for setting up an SMB cluster over Content Software for File filesystems. See the the **`weka smb cluster`** command in the *Hitachi Content Software for File Command Line Reference Guide*.

Work flow

To configure the Content Software for File SMB support, you can use either the Content Software for File system GUI or CLI commands.

1. Configure SMB cluster: Set the Content Software for File system hosts that participate in the SMB cluster.
2. Join the SMB cluster in the Active Directory: Connect and define the Content Software for File system in the Active Directory.
3. Create shares and their folders, and set permissions. By default, the filesystem permissions are `root/root/755` and initially can only be set using a Content Software for File FS/NFS mount.

Once these steps are done, it is possible to connect as an administrator and define permissions through the Windows operating system.

Establishing an SMB cluster

Before you begin

Each Content Software for File cluster only supports a single SMB cluster.

Verify that the DNS "nameserver" of the hosts participating in the SMB cluster is configured to the Active Directory server.

Each Content Software for File cluster only supports a single SMB cluster.

Procedure

1. Select the Content Software for File hosts participating in the SMB cluster and set the domain name.
2. In on-premises deployments, it is possible to configure a list of public IP addresses distributed across the SMB cluster. If a node fails, the IP addresses from that node are reassigned to another node.

Configuring the round-robin DNS server

To ensure that the various SMB clients will balance the load on the various Content Software for File hosts serving SMB, it is recommended to define a Round-robin DNS entry which will resolve to the list of floating IPs, ensuring that client loads will be equally distributed across all hosts.

Note: Make sure to set the TTL (Time to Live) for all A records assigned to the SMB servers to 0 (Zero). This ensures that the client or the DNS server does not cache the IP.

Creating SMB shares

After establishing an SMB cluster, it is possible to declare SMB shares. Each share should have a name and a share path, specifically the path into the Content Software for File filesystem, which can be the root of the filesystem or a subdirectory. This is created in the shell using either a WekaFS mount or an NFS mount.

If the share uses the root, it is not necessary to create a root folder (it already exists). If the share is declared without providing a sub-directory, the WekaFS root will be used. If sub-folders have to be created (an operation that is performed manually), the permissions have to be adjusted accordingly.

Filesystem permissions and access rights

Once the SMB cluster is connected to the Active Directory, it can assign permissions and access rights of SMB cluster filesystems to specific users or user groups. This is performed according to POSIX permissions (Windows permissions are stored in the POSIX permissions system). Any change in the Windows permissions is adapted to the POSIX permissions.

Note: The initial set of POSIX permissions is done by the user through the driver/ NFS.

Note: To obtain root access to the SMB shares, assign an Active Directory user with `uidNumber` and `gidNumber` of zero (0).

Integration with previous versions of Windows

Creating snapshots of the Content Software for File filesystem and naming the access point in the `@GMT_%Y.%m.%d-%H.%M.%S` format will expose those to the windows previous versions mechanism.

To view a list of available previous versions that correspond to the filesystem snapshots, right-click a file or a folder in the Content Software for File SMB share in the windows client, and select Properties -> Previous Versions.

For example, creating a snapshot using the CLI:

```
$ weka fs snapshot create fs_name snapshot_name --access-point `TZ=GMT date +@GMT-%Y.%m.%d-%H.%M.%S`
```

For more information, See Snapshots (on page 77) and Creating a snapshot (on page 79).

SMB management using the GUI

SMB management is described that includes the setting up an SMB cluster over Content Software for File filesystems and managing the cluster itself using the GUI.

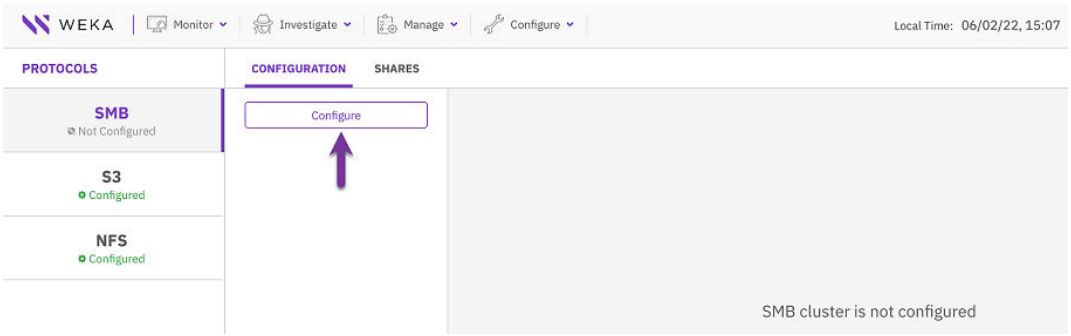
Note: Use ASCII format when configuring name fields (for example, domain, shares, among others.)

Configuring an SMB cluster using the GUI

Procedure

1. From the menu, select **Manage > Protocols**.

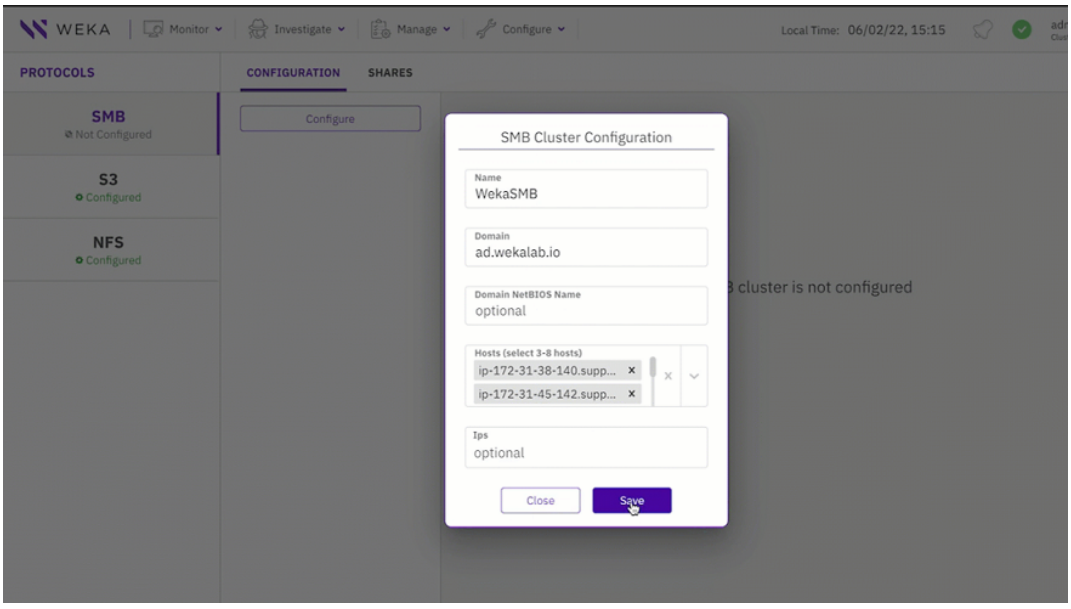
2. From the Protocols pane, select **SMB**.
3. On the SMB tab, select **Configure**.



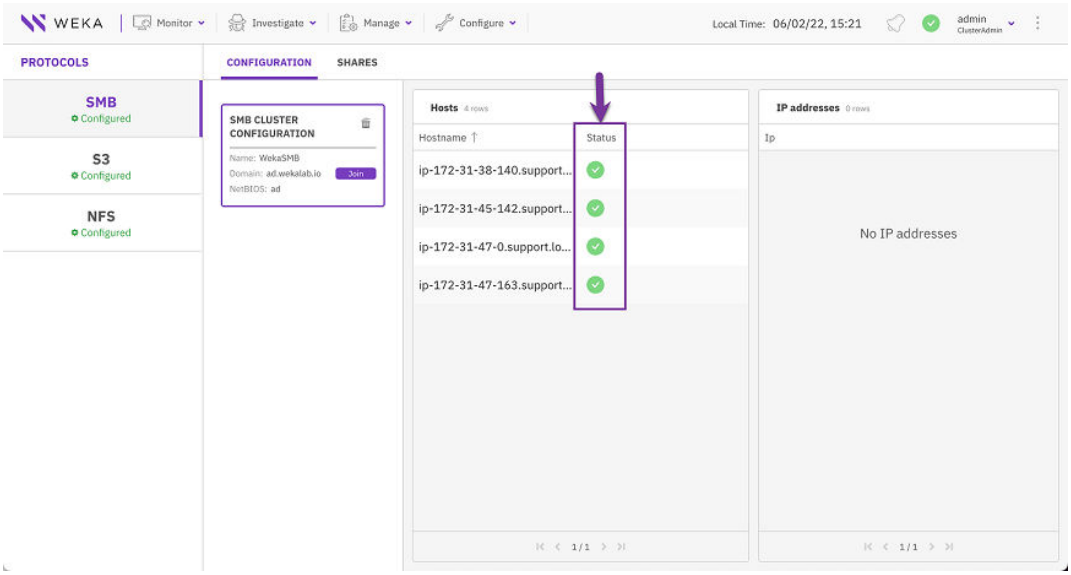
4. In the SMB Cluster Configuration dialog, set the following properties:
   - **Name**: A NetBIOS name for the SMB cluster.
   - **Domain**: The domain which the SMB cluster is to join.
   - **Domain NetBIOS Name**: (Optional) The domain NetBIOS name.
   - **Hosts**: List of 3-8 Content Software for File system hosts to participate in the SMB cluster, based on the host IDs in Content Software for File.
   - **IPs**: (Optional) List of public IPs (comma-separated) used as floating IPs for the SMB cluster to serve the SMB over and thereby provide HA (do not assign these IPs to any host on the network). For IP range, use the following format: a.b.c.x-y.

   Note: In AWS installations, it is not possible to set a list of SMB service addresses. The SMB service must be accessed using the primary addresses of the cluster nodes.

5. Select **Save**.



Once the system completes the configuration process, the host statuses change from not ready (red X icon) to ready (green V icon), as shown in the following example:



Joining the SMB cluster to an Active Directory using the GUI

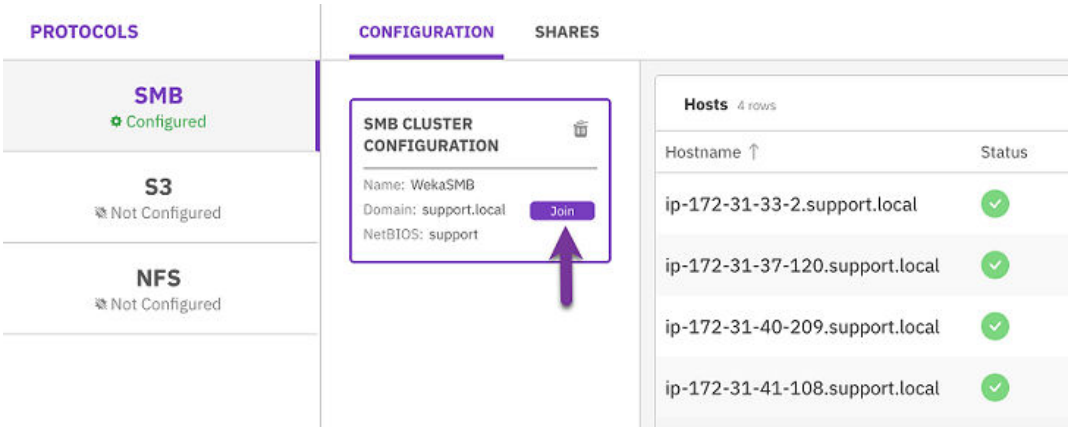
Before you begin

To enable the organizational Active Directory to resolve the access of users and user groups to the SMB cluster, join the SMB cluster in the Active Directory (AD).

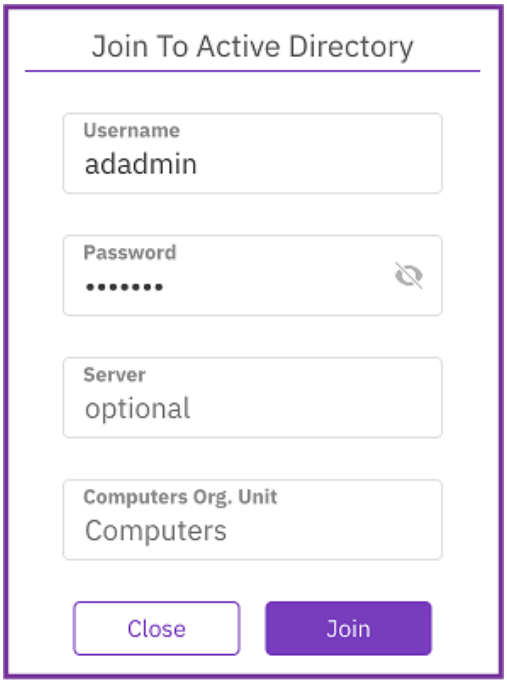
To enable the Content Software for File storage nodes to join the AD domain, verify that the AD server is the DNS server.

Procedure

1. To join the SMB cluster to an Active Directory, click the **Join** button when all hosts have been prepared and are ready. The following window will be displayed:



2. In the Join to Active Directory dialog, set the following properties:
   - **Username** and **Password**: A username and password of an account that has access privileges to the Active Directory. Content Software for File does not save the user password. A computer account is created on behalf of the user for the SMB cluster.
   - **Server**: (Optional) Content Software for File identifies the AD server automatically based on the AD name. You do not need to set the server name. In some cases, if required, specify the AD server.
   - **Computers Org. Unit**: The default organization unit is the Computers directory. You can define any other directory to connect to in Active Directory, such as SMB servers or Corporate computers.



Once the SMB cluster joins in the Active Directory, the join status next to the domain changes to **Joined**.

Note: To join a different Active Directory to the existing SMB cluster configuration, select **Leave**. To confirm the action, enter the username and password used to connect to the Active Directory.

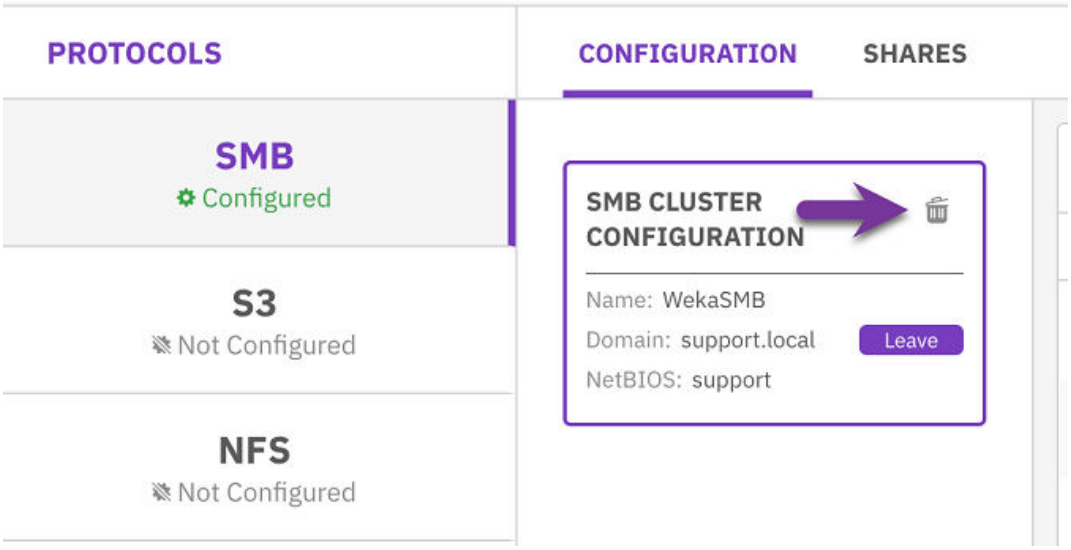
Deleting an SMB cluster using the GUI

Before you begin

Deleting the SMB cluster resets its configuration data.

Procedure

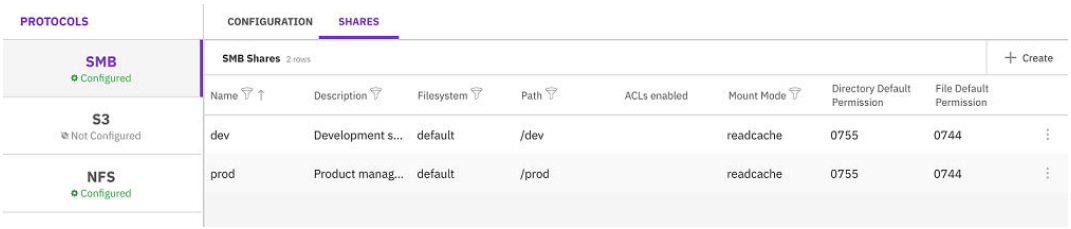
1. In the SMB Cluster Configuration, select the **trash** icon.
2. In the SMB Configuration Reset message, select **Reset**.



Displaying the SMB shares list using the GUI

Procedure

1. From the menu, select **Manage > Protocols**.
2. From the Protocols pane, select **SMB**.
3. Select the **Shares** tab. You can filter the list using any column in the table.



Adding an SMB share using the GUI

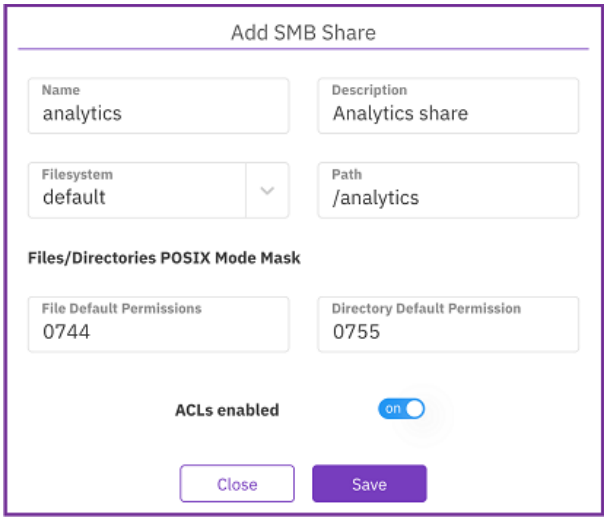
Procedure

1. In the Shares tab, select **+Create**.

2. In the Add SMB Share dialog, set the following properties:
   - **Name**: A meaningful name for the SMB share.
   - **Description**: A description of the SMB share.

- **Filesystem**: The filesystem to use for the SMB share. Select one from the list. A filesystem set with required authentication cannot be used for SMB share.
- **Path**: A valid internal path, relative to the root, within the filesystem to expose for the SMB share.
- **Files/Directories POSIX Mode Mask**: Set the new default file and directory permissions in a numeric (octal) format created through the share.
- **ACLs Enabled**: Determines whether to enable the Windows Access-Control Lists (ACLs) on the share. Weka translates the ACLs to POSIX.

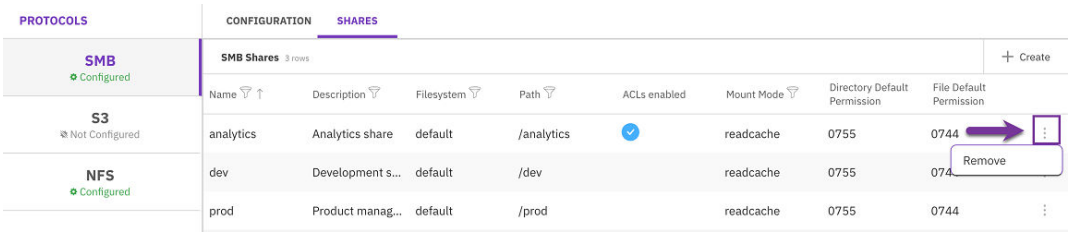
3. Select **Save**.



Removing an SMB share using the GUI

Procedure

1. In the Shares tab, select the three dots of the share and select Remove.



2. In the confirmation message that appears, select Confirm. The removed share no longer appears in the SMB Shares list.

Chapter 13: Alerts

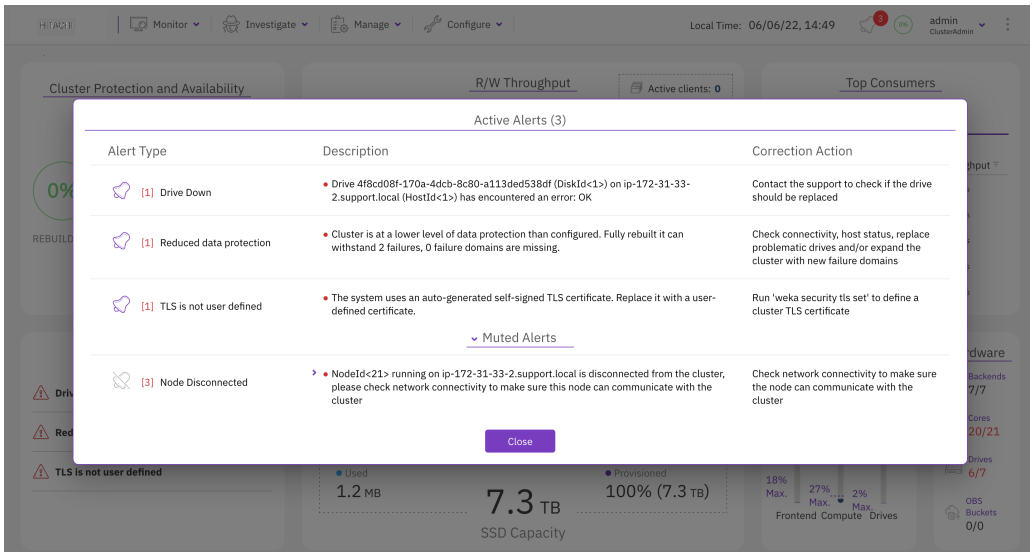
The alerts that can be received in this version of the Content Software for File system are described.

Overview

Alerts indicate problematic ongoing states that the cluster is suffering from. To dismiss an alert, you need to resolve the root cause of the alert.

For each alert, the system provides the alert name, its description, and the corrective action.

Usually, an alert is introduced alongside an equivalent event. This can help in identifying the point in time that the problematic state occurred and its root cause.



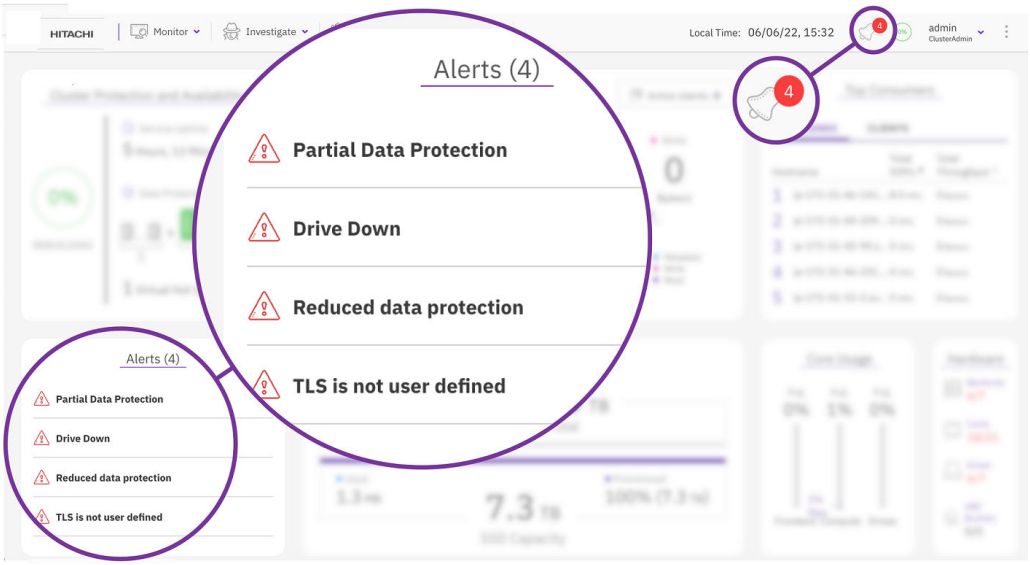
Manage alerts using the GUI

How to manage alerts using the GUI.

Viewing alerts using the GUI

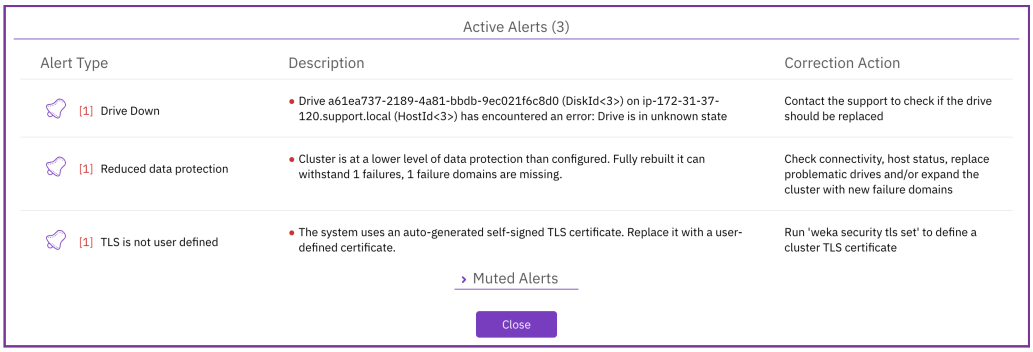
The bell icon on the top bar indicates the number of the existing active alerts in the system. The alerts pane in the system dashboard also provides the name of the alerts.

If there are no alerts (active or muted), the alerts pane is empty, and the bell does not specify any number.



Procedure

1. To display the alert details, select the bell icon or select any alert.



Muting alerts

Before you begin

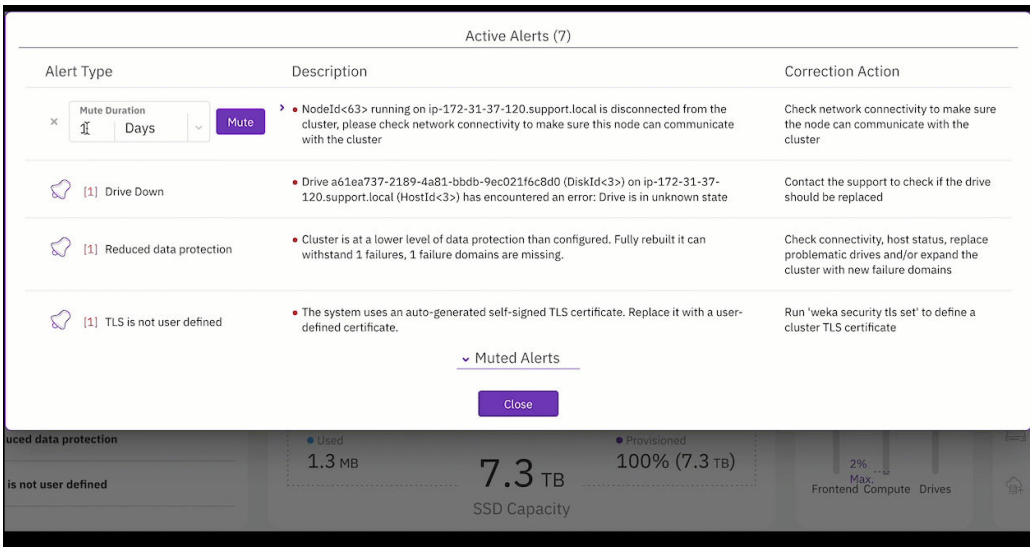
If for any reason, it is not possible to resolve the root cause of an alert in a reasonable time and you want to hide it temporarily, you can mute the alert for a specified period. Then later, you can unmute the alert and resolve it.

The system automatically unmutes the muted alerts after the expiry period.

Procedure

1. On the Active Alerts page, select the bell next to the alert.
2. Set the mute duration (number and units) and select **Mute**.

   The muted alert is moved to the Muted Alerts area. The total number of active alerts is deducted by the number of muted alerts.

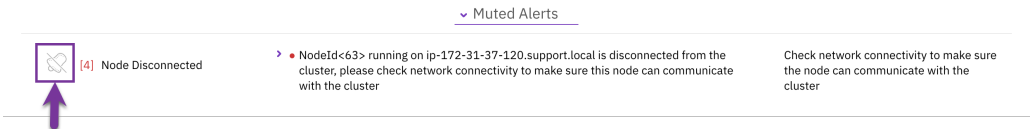


Unmute alerts

Muted alerts appear under the Muted Alerts area. You can unmute an alert manually before the expiry duration.

Procedure

1. Under the Muted Alerts area, select the bell of the alert you want to unmute.



List of alerts

Name	Description	Actions
AdminDefault Password	The admin password is still set to the factory default.	Change the admin user password to ensure only authorized users can access the cluster.
AgentNotRunning	The Content Software for File local control agent is not running on a host.	Restart the agent with service Content Software for File-agent start.
ApproachingClientsUnavailability	Approaching the maximum amount of clients that can connect with the current cluster resources.	Make sure all backhand servers are up or expand the cluster with more backend servers.
AutoRemoveTimeoutTooLow	Stateless Client auto-remove timeout too low.	Remount the host with a higher auto-remove timeout value.
BackendNumaBalancingEnabled	A host has automatic NUMA balancing enabled which can negatively impact performance.	To disable, run echo 0 > /proc/sys/kernel/numa_balancing on the backend host.
BackendVersionsMismatch	There are mismatching versions of backend servers in the cluster.	Upgrade all the backend servers to match the cluster's version.
BondInterfaceCompromised	The host is configured to work with a highly available network, but has lost the connectivity redundancy. A single network failure can disconnect the host from the cluster, which will result in the unavailability of data to the host (in case of a client host) or data protection reduced redundancy (in case of a backend host).	Check the network configuration, cables, NICs to resolve the issue.

Name	Description	Actions
BucketHasNoQuorum	Too many compute nodes are down, causing the bucket compute resource to be unavailable.	Check that the compute nodes and their hosts are up and running and fully connected; Contact customer support if the issue is not resolved.
BucketUnresponsive	A compute resource has failed, causing system unavailability.	Check that the compute nodes and their hosts are up and running and fully connected; Contact customer support if the issue is not resolved.
ChokingDetected	High congestion level detected in the cluster.	For more information, refer to System Congestion (on page 205).
ClientNumaBalancingEnabled	A host has automatic NUMA balancing enabled which can negatively impact performance.	To disable, run `echo 0 > /proc/sys/kernel/numa_balancing` on the client host.
ClientVersionsMismatch	There are clients with a version that does not match the cluster version. Some features may not be available until all the clients are upgraded.	Upgrade clients to be in the same version as the cluster by locally running `weka local upgrade`.
ClockSkew	The clock of a host is skewed in relation to the cluster leader, with a time difference more than the permitted maximum of 30 seconds.	Make sure NTP is configured correctly on the hosts and that their dates are synchronized.
CloudHealth	A host cannot upload events to the Content Software for File cloud.	Check the host has Internet connectivity and is connected. For

Name	Description	Actions
		details, contact your Hitachi representative.
CloudStatsError	Statistics upload to Content Software for File cloud failed.	Check the host has Internet connectivity and is connected to the Content Software for File cloud as explained in the
ClusterInitializationError	The cluster has encountered an error while initializing.	Fix the underlying problem causing the error to successfully start IO operations.
ClusterIsUpgrading	Cluster is upgrading.	If the upgrade doesn't finish normally, contact customer support for assistance.
CPUFrequentStarvation	CPU frequent starvation detected in the last minute.	Check the relevant hosts logs for potential hardware problems or core allocation issues.
CPUStarvation	Content Software for File processes are experiencing long CPU stalls.	Check the relevant hosts logs for potential hardware problems.
DataIntegrity	Data integrity issue found.	Contact customer support.
DataProtection	Some of the system's data is not fully redundant.	Check which node/ host/drive is down and act accordingly.
DedicatedWatchdog	A dedicated Content Software for File host requires the installation of a watchdog driver. Make sure a watchdog is available at /dev/watchdog.	For more information, contact customer support.
DriveDown	A drive is not responding.	Contact customer support to check if

Name	Description	Actions
		the drive should be replaced.
DriveEndurancePercentageUsed	Drive exceeding its life expectancy.	It is recommended to replace the drive before it fails.
DriveEnduranceSparesRemaining	Drive internal spares running too low.	It is recommended to replace the drive before it fails.
DriveNeedsPhaseout	A drive has too many errors.	Phase-out the drive and probably replace it.
FilesystemHasTooManyFiles	The filesystem storage configuration for the size of file and directory entries is exceeding (or about to exceed).	Increase the max-files for the filesystem.
FilesystemSquashPending	A filesystem squash task is pending.	The filesystem is pending squash. The squash background task begins automatically. No corrective action is required.
FilesystemsThinProvisioningLowSpace	There are thinly provisioned filesystems that running on low free capacity.	Consider adding more SSD capacity to the organization containing these filesystems'.
FilesystemsThinProvisioningReserveReached	The request reserved capacity (for filesystem creation/expansion) is available.	The reserved capacity can now be used for filesystems creation/ expansion.
HangingCacheSync	Cache sync is stopped	A stopped cache sync can prevent other clients from accessing some files. To resolve this issue, reboot the host or remove it from the cluster.Data that is

Name	Description	Actions
		not synced with the cluster may be lost.
HangingIOs	Some IOs are hanging on the node acting as a driver/NFS/backend.	Check that the compute nodes and their hosts are up and running, and fully connected. Also check that if a backend object store is configured, it is connected and responsive. Contact customer support if the issue is not resolved.
HighDrivesCapacity	The average capacity of the SSDs is too high.	Free-up space on the SSDs or add more SSDs to the cluster.
HighLevelOfUnreclaimedCapacityInObjectStore	High level of unreclaimed space in object store.	Check object store connectivity and deletion operations' progress. Validate authorization of deletion operations on the object store. Run `weka fs tier capacity` for details.
JumboConnectivity	A host cannot send jumbo frames to any of its cluster peers.	Check the host network settings and the switch to which it is connected, even if Content Software for File seems to be functional since this will improve performance.
KmsError	KMS Error.	Review the KMS credentials, permissions, and

Name	Description	Actions
		configuration, as suggested in KMS management (on page 209).
LicenseError	A license conflict exists.	Make sure the cluster is using a correct license, the license has not expired, and the cluster allocated space does not exceed the license.
LowDiskSpace	The host has low disk space (for `/opt/weka` directory) which can affect some Content Software for File reporting services.	Free up space on the host, or contact customer support
ManualOverridesActive	Manual overrides are active.	Please contact customer support
MismatchedDriveFailureDomain	The drive failure domain does not match the failure domain of its attached host.	Either connect the mismatched drive to a host with a matching failure domain, or re-provision the drive to erase its failure domain.
NegativeUnprovisionedCapacity	Content Software for File capacity usage changes detected due to cluster upgrade.	One or more of the filesystems need to be resized in order to reclaim capacity. Contact customer support.
NetworkInterfaceLinkDown	A Network interface has a link down status.	Check the connectivity to the interface and see if there is a blocking it.
NoClusterLicense	No license is assigned to the cluster.	Obtain and install a license from customer support.
NodeBlacklisted	There is a blacklisted node in the cluster.	Use Content Software for File

Name	Description	Actions
		`debug blacklist disable` to whitelist nodes so they can rejoin the cluster.
NodeDisconnected	A node is disconnected from the cluster.	Check network connectivity to make sure the node can communicate with the cluster.
NodeNetworkUnstable	A node seems to have an unstable network. As a consequence, it has been fenced by the system and does not contribute resources to the Content Software for File cluster.	Make sure there is no network connectivity issue in the cluster. Contact customer support if the issue is not resolved.
NodeRDMANotActive	RDMA is supported on the host but it is inactive.	Make sure Mellanox OFED version 4.6 or higher is properly installed on the host.
NodeTieringConnectivity	A node cannot connect to an object-store.	Check connectivity with the object store and make sure the node can communicate with it.
NotEnoughActiveDrives	There are not enough active failure domains.	Check connectivity, host status, and/or replace problematic drives.
OFEDVersions	A host Mellanox OFED version ID does not match the one used by the Content Software for File container.	Install a supported OFED. If the current version needs to be retained or the alert continues after a supported version is installed,

Name	Description	Actions
		contact customer support.
PartialConnectivityTrackingDisabled	The cluster's partial connectivity tracking mechanism is disabled, affecting the cluster's self-healing capabilities.	Contact customer support.
PartiallyConnectedNode	A node seems to be only partially connected.	Make sure there is no network connectivity issue. Contact customer support if the issue is not resolved.
PassedClientsAvailabilityThreshold	Reached Clients Limit	Add more backend servers to the cluster, check whether backends are down, or disconnect some clients.
PerformanceDegradedLowRAM	The host is running low on RAM. Additional Metadata entries are swapped to the SSD. This might impact performance.	Make sure all the compute hosts and processes are up, add more hosts to the Content Software for File cluster, or the configured RAM of the cluster backend hosts.
QuotasHardLimitReached	There are directory quotas that have reached their hard limit.	Run `weka fs quota list` to see which directory quotas have reached their hard limit.
QuotasSoftLimitReached	There are directory quotas that have reached their soft limit.	Run `weka fs quota list` to see which directory quotas have reached their soft limit.

Name	Description	Actions
ResourcesNotApplied	There are changes to host resources that are not applied in the Content Software for File cluster.	To apply changes run Content Software for File cluster `host apply <host_id>`
SSDCapacityDiscrepancy	Used SSD capacity mismatches the expected range	Monitor COMPUTE processes' stability, contact customer support.
SystemDefinedTLS	The Content Software for File cluster uses an auto-generated self-signed certificate.	Run `weka security tls set` to replace the auto-generated certificate with your own certificate for cluster TLS use.
TLSCertificateExpired	TLS Certificate has expired.	Replace the current certificate using Content Software for File `security server-tls set`.
TLSCertificateExpiresSoon	TLS Certificate is about to expire.	Replace the current certificate using Content Software for File `security server-tls set`.
TieredFilesystemOverfillingSSD	Tiered filesystems' SSD Capacity overfilling.	Resolve tiering connectivity issues or increase the upload bandwidth.
TraceDumperDown	Trace dumper is down	Contact customer supportto restart the trace dumper.
TracesDisabled	Traces are disabled.	To turn them back on contact customer support.

Name	Description	Actions
TracesFreezePeriodActive	A trace freeze period is active.	Some traces can be protected from rotating for a period of time to debug the system. This is done by the customer support when needed. If the issue persists after the case has been resolved please contact customer support.
UdpModePerformanceWarning	The backend host is configured in UDP mode.	If this is a misconfiguration use Content Software for File `cluster host net add` to add network devices to this host.

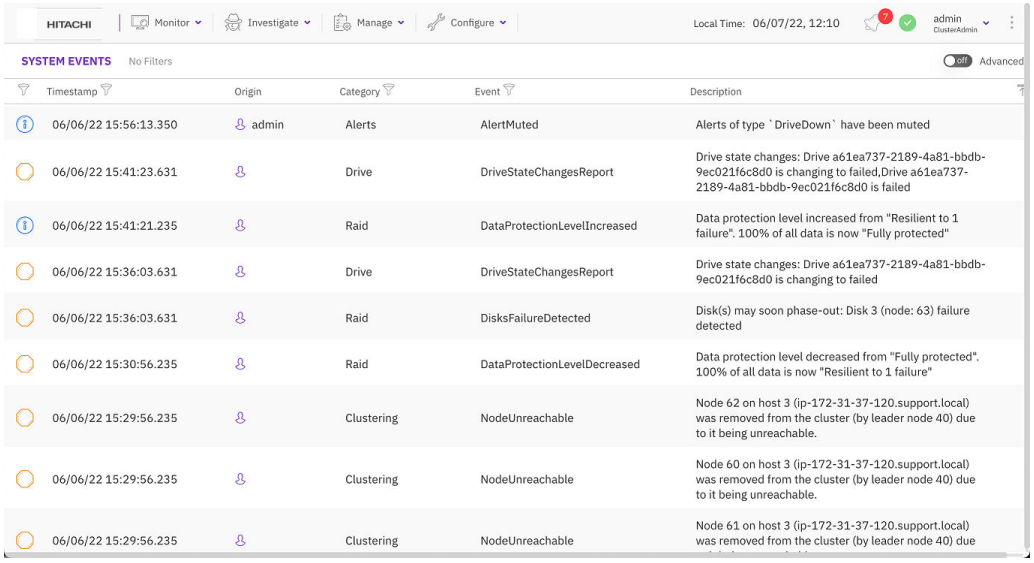
Chapter 14: Events

The events available in the Content Software for File system and how to work with them is described.

Overview

Content Software for File events indicate relevant information concerning the Weka cluster and customer environment. Triggered by a customer operation or an environment change, events can be informational, indicate an issue in the system, or indicate a previously resolved issue.

The Content Software for File system sends all events to a predefined central monitoring system.



Managing events using the GUI

How to manage events using the GUI.

Viewing events using the GUI

The events enable you to investigate issues that occur in the system.

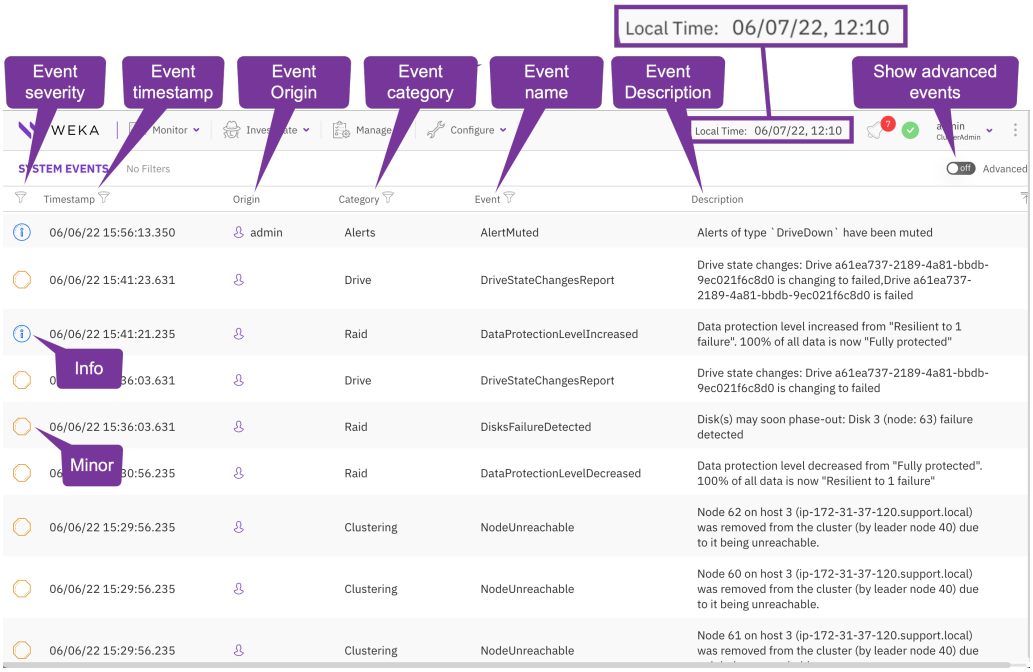
The System Events page provides the following details:

- Severity: The severity of the event. The options are Info (lowest), Warning, Minor, Major, and Critical (highest).
- Timestamp: The date and time the event occurred. You can switch the display time between local and system time through the top bar.
- Origin: The originator of the event. For example, when a user creates a filesystem, the username appears as the event's originator.
- Category: The category options include Alerts, Cloud, Clustering, Drive, Events, Filesystem, IO, InterfaceGroup, Licensing, NFS, Network, Node, ObjectStorage, Raid, Statistics, System, Upgrade, and User.
- Name: The name of the event.
- Description: The description of the event.

Tip: You can select the Advanced switch to display internal events. This option is helpful for experts investigating internal issues.

Procedure

1. From the menu, select **Investigate** > **Events**.



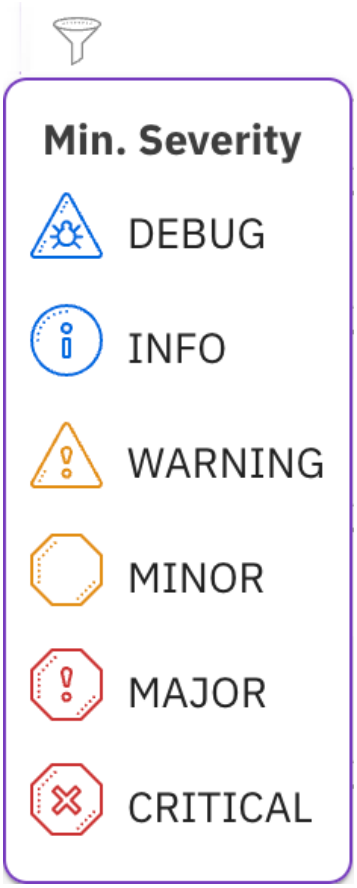
Filtering events using the GUI

You can filter the events according to the event severity, timestamp, category, or event name. You can also filter events by multiple categories and multiple event names.

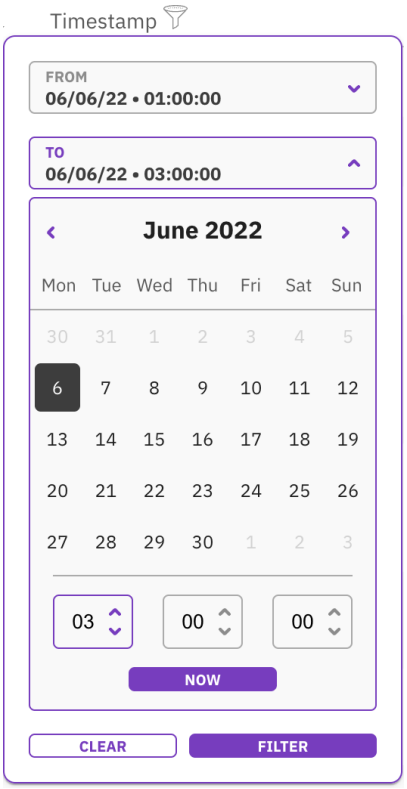
Procedure

1. To display events of a specific minimum severity:
   - Select the filter icon of the **Severity** column.
   - Select the required minimum severity.

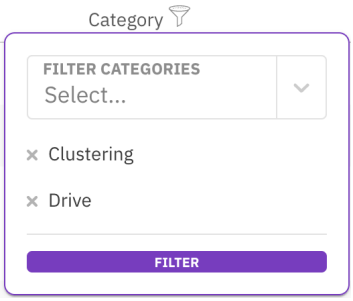
   For example, if you select the Major severity, also the Critical events are displayed.



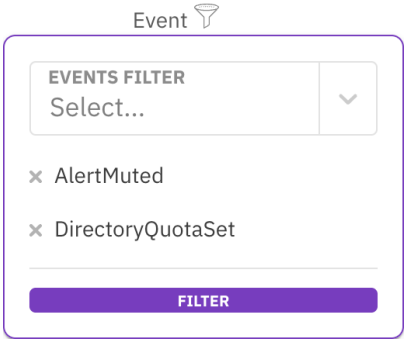
2. To display events that occur during a specific period:
   - Select the filter icon of the **Timestamp** column.
   - In the **From** field, select the timestamp of the beginning of the period to display.
   - In the **To** field, select the timestamp of the end of the period to display, or select Now.
   - Select **Filter**.
   -



3. To display events of specific categories:
   - Select the filter icon of the **Category** column.
   - In the **Filter Categories**, select the category you want to display. You can select multiple categories.
   - Select **Filter**.



4. To display events with specific event names:
   - Select the filter icon of the **Event** column.
   - In the **Events Filter**, select the event name you want to display. You can select multiple event names.
   - Select **Filter**.



List of events

All the events generated by the Content Software for File system, according to Category are listed.

Alerts

Type	Severity	Description
AlertMuted	INFO	Alert muted
AlertUnmuted	INFO	Alert unmuted

Cloud

Type	Severity	Description
CloudDisabled	INFO	Cloud disabled
CloudEnabled	INFO	Cloud enabled
CloudProxyUpdated	INFO	Cloud proxy updated
CloudSetUploadRate	INFO	Cloud upload rate changed
CloudStatsErrorClearedEvent	WARNING	Cloud stats have now been written successfully
CloudStatsErrorEvent	WARNING	Error writing cloud stats for upload
DiagsUploaded	INFO	Diags uploaded
LowDiskSpaceClearedEvent	WARNING	Host no longer has low disk space

Type	Severity	Description
LowDiskSpaceEvent	WARNING	Host has low disk space

Clustering

Type	Severity	Description
AllBucketsResponsive	INFO	All compute resources are now responding.
BucketRedist	INFO	Buckets were redistributed in the cluster.
ClientConnected	INFO	Client connected.
ClientDisconnected	INFO	Client disconnected.
ClientNodesFencedDuringStopIO	INFO	Some clients disconnected during stop-io
ClientRemoved	INFO	Disconnected client is being removed from the cluster.
ClientsUnavailable	CRITICAL	Some clients are unavailable because too many backends are down.
ClockSkewedHostJoin	MINOR	Host cannot join because of clock skew
ClusterInitializationFailed	MAJOR	Cluster initialization failed.
ClusterInitialized	INFO	Cluster successfully initialization.
ClusteringFailure	MINOR	Node clustering failed.
ConfigChangeSetsSliderFull	MINOR	Config changeset slider is full while the node is pulling config.
ConfigGenerationHasNoFirstChunk	MINOR	Applying a partial config generation is prohibited
ConfigSnapshotPulled	MINOR	Config snapshot pulled.
GrimReaperFencingNode	MINOR	Partially connected node selected to be fenced by grim reaper.

Type	Severity	Description
HostActivated	INFO	Host configuration change.
HostAdded	INFO	Host configuration change.
HostAdding	INFO	Host configuration change.
HostDeactivated	INFO	Host configuration change.
HostDeactivating	INFO	Host configuration change.
HostRemoved	INFO	Host configuration change.
HostRemovingFailed	INFO	Host configuration change.
HostRemoving	INFO	Host configuration change.
LeaderChanged	WARNING	Cluster leader has changed.
NodeNetworkUnstable	MAJOR	A node with unstable network detected.
NodePartiallyConnected	MINOR	A partially connected node was removed.
NodeRejoined	INFO	Node rejoined the cluster.
NodeUnreachable	MINOR	An unreachable node was removed.
PreviousCluster	INFO	This host was part of another cluster before.
RejoinFailureReport	MINOR	Node(s) failed to rejoin.
UnresponsiveBuckets	CRITICAL	Some compute resources are not responding.
WrongConfigSignatureForRaftSnapshot	MINOR	Tried loading RAFT snapshot with unsupported config root snapshot signature
WrongSchemaVersionForRaftSnapshot	MINOR	Tried loading RAFT snapshot with unsupported schema version

Config

Type	Severity	Description
DirectoryQuotasDisabled	INFO	Directory Quotas were disabled
DirectoryQuotasEnabled	INFO	Directory Quotas were enabled
LoginBannerCleared	INFO	Login banner has been cleared
LoginBannerDisabled	INFO	Login banner disabled
LoginBannerEnabled	INFO	Login banner enabled
LoginBannerSet	INFO	Login banner has been set
S3ClusterCreated	INFO	S3 Cluster Created
S3ClusterDestroyed	INFO	S3 Cluster Destroyed
S3ClusterUpdated	INFO	S3 Cluster Updated

Custom

Type	Severity	Description
Custom	INFO	Custom event.

Drive

Type	Severity	Description
CorruptedDrive	MAJOR	Drive has a valid header but is corrupt.
DriveAdded	INFO	Drive provisioned.
DriveDeactivated	INFO	Drive deactivated.
DriveExcessiveErrors	WARNING	Drive has excessive error rate and will be phased out; call Contact Support.
DriveFormatUpgraded	INFO	Drive format was upgraded.

Type	Severity	Description
DriveImmediateShutdown	MAJOR	Drive had to be shutdown immediately; call Contact Support.
DriveInfoReport	INFO	Drive Information reporting.
DriveInitFailed	MAJOR	Drive failed to initialize.
DriveIoErrorBMS	MAJOR	Drive found an IO error in background media scan.
DriveIoError	MAJOR	Drive had an IO error.
DriveLimitExceeded	WARNING	Attempted to add more drives than supported.
DriveMediumError	MINOR	Drive had a Medium error.
DriveNvmeErrorLog	WARNING	NVMe Drive Error Log Entry.
DriveNvmeSmartChange	MINOR	NVMe Drive SMART status changed.
DriveNvmeSmartInfo	INFO	NVMe Drive SMART status update - drive normal.
DriveRemoved	INFO	Drive removed.
DriveSignatureUnknown	MINOR	Drive has an unknown signature.
DriveSmartCriticalWarning	MINOR	Drive SMART reports critical warning, failing it immediately
DriveStateChangesReport	MINOR	Drive state changes.
DriveTrimAborted	WARNING	Drive TRIM commands at initialization timed out and aborted, potentially degrading write performance
DriveUnderIOMMU	MAJOR	Drive is under IOMMU and cannot be used.
DriveUnresponsive	MAJOR	Drive is unresponsive and failed to return IOs for an extended period of time; consider power cycling the host.

Type	Severity	Description
DriveWrongFailureDomain	MINOR	Drive is attached to a host from an incorrect failure domain.
NvmeBindTimingOut	MAJOR	NVMe device bind is stuck, server needs power cycle to recover.

Events

Type	Severity	Description
DedupEventsDiscarded	WARNING	Deduplicated events discarded.
EventsDedupReport	INFO	Event deduplication ended.
EventsDiscarded	MINOR	Too many events were generated in a short period of time, so some of them were discarded and lost.
ExampleAggregated	INFO	Example Aggregated.
ExampleDebug	DEBUG	ExampleDebug.
Example	INFO	Example.
TracesDumperDownEvent	MAJOR	Traces Dumper is inactive

Filesystem

Type	Severity	Description
BlockReadFailure	CRITICAL	Failed to read a block.
BlockSeekFinished	MAJOR	Block seek finished
BlockSeekStarted	MAJOR	Block seek started for a secondary metadata block that could not be read
BrokenFile	MAJOR	File metadata corruption
CWTaskTemplateFinished	INFO	CWTask template finished

Type	Severity	Description
CacheFlushHanging	MAJOR	Host is hanging while trying to sync a file's write cache to the cluster
ChecksumErrorInCommit	MAJOR	Checksum error detected by SSD node in a committed block.
ChecksumErrorInWrite	CRITICAL	Checksum error detected by COMPUTE node in a write
DefaultDirectoryQuotaSet	INFO	Default directory quota was set
DefaultDirectoryQuotaUnset	INFO	Default directory quota was unset
DirectoryQuotaSet	INFO	Directory quota was set
DirectoryQuotaUnset	INFO	Directory quota was unset
DumpSnapHashCompleted	INFO	Finished a snap hash manifest scan
FailedToSplitSliceNoRetry	CRITICAL	Failed to split a directory slice - wont retry.
FilesystemAdded	INFO	Filesystem configuration change.
FilesystemDeleted	INFO	Filesystem configuration change.
FilesystemDownloadStarted	INFO	Filesystem download started.
FilesystemGroupAdded	INFO	Filesystem group configuration change.
FilesystemGroupDeleted	INFO	Filesystem group configuration change.
FilesystemGroupUpdated	INFO	Filesystem group configuration change.
FilesystemRemoved	INFO	Filesystem configuration change
FilesystemSquashFinished	INFO	
FilesystemSquashStarted	INFO	

Type	Severity	Description
FilesystemUpdated	INFO	Filesystem configuration change.
ForcedBucketStepdown	MINOR	Bucket forced to step down.
FsCapacityLimitReached	WARNING	Filesystem capacity limit has been reached
HangingBackendIosDetected	CRITICAL	Some IOs are hanging.
HangingBackendIosNoLongerDetected	INFO	IOs are no longer hanging.
HangingBucketStepDown	WARNING	Bucket step-down is hanging.
HangingDirectorySplit	CRITICAL	Directory split hasn't any made progress for a long time.
HangingDriverFrontendIosDetected	CRITICAL	Some IOs are hanging.
HangingDriverFrontendIosNoLongerDetected	INFO	IOs are no longer hanging.
HangingNFSFrontendIosDetected	CRITICAL	Some IOs are hanging.
HangingNFSFrontendIosNoLongerDetected	INFO	IOs are no longer hanging.
IntegrityCheckFinished	INFO	Integrity check finished
IntegrityCheckIssue	CRITICAL	Found an data integrity issue
IntegrityCheckStarted	INFO	Integrity check started
IntegrityCheckTransientIssue	WARNING	Found a possibly transitional data integrity issue - check if a critical issue is found afterwards
ManualOverrideStall	WARNING	Service has been manually-overridden and stalled
ObjectStorageAttachedToFilesystem	INFO	Object Storage attached to filesystem.
ObjectStorageFinishedDetachingdFromFilesystem	INFO	Object Storage finished detaching from filesystem.

Type	Severity	Description
ObjectStorageStartedDetachingdFromFilesystem	INFO	Object Storage started detaching from filesystem.
QuotaGraceExpired	WARNING	Directory soft capacity quota has been reached and grace period expired
QuotaHardLimitReached	WARNING	Directory hard capacity quota has been reached
RAIDDataBlockReadFailureInSnaphashDump	WARNING	Failed to read data block from RAID when dumping the snapshot manifest.
RAIDMDReadFailureInSnaphashDump	WARNING	Failed to read metadata block from RAID when dumping the snapshot manifest.
SnapshotContentCopied	INFO	Snapshot content copied.
SnapshotCreated	INFO	Snapshot created.
SnapshotDeleted	INFO	Snapshot deleted.
SnapshotDownloadStarted	INFO	Snapshot download started.
SnapshotFilesystemRestored	INFO	Filesystem restored from snapshot.
SnapshotParamsUpdated	INFO	Snapshot updated.
SnapshotUploadFinished	INFO	Snapshot upload finished
SnapshotUploadStarted	INFO	Snapshot upload started.
SquelchBlockIdSetAbortedFlushed	DEBUG	While setting a squelch block's block id for upgrade was already changed to invalid
SquelchBlockIdSetAbortedRewritten	WARNING	While setting a squelch block's block id for upgrade was already rewritten to something else
SuperblockUnreadable	CRITICAL	Superblock of a bucket could not be loaded
UnflushedOpOnDeletingSnapview	MAJOR	Unflushed IO on a deleting snapshot.

IO

Type	Severity	Description
SystemDriveIsTooSlow	MAJOR	System drive is slow to respond.

InterfaceGroup

Type	Severity	Description
FloatingIpAcquired	INFO	Floating IP was acquired by Node.
FloatingIpReleased	INFO	Floating IP was released by Node.
InterfaceGroupAdded	INFO	Interface group configuration change.
InterfaceGroupDeleted	INFO	Interface group configuration change.
InterfaceGroupIpsAdded	INFO	Interface group IPs configuration change.
InterfaceGroupIpsDeleted	INFO	Interface group IPs configuration change.
InterfaceGroupPortAdded	INFO	Interface group port configuration change.
InterfaceGroupPortDeleted	INFO	Interface group port configuration change.
InterfaceGroupUpdated	INFO	Interface group configuration change.

KMS

Type	Severity	Description
KmsConfigurationAdded	INFO	KMS configuration configuration change.
KmsConfigurationRemoved	INFO	KMS configuration configuration change.

Type	Severity	Description
KmsConfigurationUpdated	INFO	KMS configuration configuration change.

Licensing

Type	Severity	Description
LicensingReset	INFO	Licensing state has been reset.
NewLicenseInstalled	INFO	New license installed.
PaygLicensingEnabled	INFO	PAYG licensing enabled.

ManualOverride

Type	Severity	Description
ManualOverrideChanged	INFO	Manual override changed

NFS

Type	Severity	Description
NfsClientGroupAdded	INFO	NFS client group configuration change.
NfsClientGroupDeleted	INFO	NFS client group configuration change.
NfsClientGroupRuleAdded	INFO	NFS client group rule configuration change.
NfsClientGroupRuleDeleted	INFO	NFS client group rule configuration change.
NfsExportsPermissionsAdded	INFO	NFS export permissions for configuration change.
NfsExportsPermissionsDeleted	INFO	NFS export permissions for configuration change.

Type	Severity	Description
NfsExportsPermissionsUpdated	INFO	NFS export permissions for configuration change.
NfsMountFail	WARNING	NFS mount request failed.
NfsPortmapFail	MAJOR	NFS server failed to register in portmap.

Network

Type	Severity	Description
ClientNodeDisconnected	INFO	Client Node disconnected from cluster
DefaultDataNetworkingChange	INFO	Default data networking configuration changed.
DpdkPoolSummary	DEBUG	Summary of DPDK pool status
HangingRPCs	MAJOR	RPCs are hanging too long.
HugepagesAllocationFailure	MINOR	Hugepages allocation failure.
IONodeCannotFetchConfig	WARNING	ode cannot join cluster for too long.
MgmtNodeCannotFetchConfig	WARNING	Node cannot join cluster for too long.
NICNotFound	INFO	NIC not found when initializing.
NetDeviceLinkDown	MINOR	Network interface DOWN.
NetDeviceLinkUp	MINOR	Network interface UP.
NetSlaveDeviceLinkDown	MAJOR	Network slave interface DOWN.
NetSlaveDeviceLinkUp	MINOR	Network slave interface UP.
NetworkPortConfigFail	MINOR	Network port configuration failed.
NetworkPortDead	MAJOR	Network Port hasn’t passed packets for a long period of time, it is likely dead.

Type	Severity	Description
NoConnectivityToLivingNode	MAJOR	Node is disconnected from living peer(s).
NoHardwareWatchdog	MAJOR	No hardware watchdog found.
NoJumboFrames	MINOR	Network does not allow large-enough messages through.
NodeCannotJoinCluster	WARNING	Node cannot join cluster for too long.
NodeCannotSendJumboFrames	MINOR	Node cannot send jumbo packets.
NodeDisconnected	MINOR	Node disconnected from cluster.
RDMAClientDisabled	MINOR	RDMA optimization disabled.
RDMAClientEnabled	MINOR	RDMA optimization enabled.

Node

Type	Severity	Description
AssertionFailed	MAJOR	Assertion failed
GCCrashReport	MINOR	Node has crashed in GC on the previous run.
NodeAbruptExitReport	MINOR	Node has crashed on the previous run.
NodeExceptionExit	MAJOR	Node exited with an exception.
NodeKernelStack	WARNING	Kernel stack of node before reset.
NodeStarted	INFO	Node started.
NodeStopped	INFO	Node stopped.
NodeTraceback	WARNING	Traceback of node before reset.

ObjectStorage

Type	Severity	Description
ChecksumErrorInDownloadedObject	CRITICAL	Checksum error detected by COMPUTE node in a downloaded OBS data block.
ChecksumErrorOnObjectUpload	MAJOR	Checksum error detected by COMPUTE node when uploading an OBS data block (corrupted after verifying data read from the drive)
DataBlobDownloadFailed	WARNING	Failed downloading data blob header.
DownloadedExtentHasInvalidBlobId	MAJOR	Downloaded extent has invalid blob id
DownloadedExtentMissingExpectedBlock	MAJOR	Downloaded extent missing expected block
InvalidDataBlobHeader	MAJOR	Invalid header detected by COMPUTE node in a downloaded OBS data blob.
ObjectStorageBucketAdded	INFO	Object storage bucket configuration change.
ObjectStorageBucketDeleted	INFO	Object storage bucket configuration change.
ObjectStorageBucketUpdated	INFO	Object storage bucket configuration change.
ObjectStoreGroupAdded	INFO	Object store configuration change
ObjectStoreGroupDeleted	INFO	Object store configuration change
ObjectStoreGroupUpdated	INFO	Object store configuration change
ObjectStoreHasHighLevelOfUnreclaimedCapacity	WARNING	Object store has high level of unreclaimed capacity
ObjectStorageIsFull	CRITICAL	Object storage is full.

Type	Severity	Description
ObjectStoreNoLongerHasHighLevelOfUnreclaimedCapacity	INFO	Object store has high level of unreclaimed capacity
ObjectStoreStatusDown	MAJOR	Object Store status is now down
ObjectStoreStatusUp	INFO	Object Store status is now up
ObsIsMissingObject	MAJOR	Permanently failed to download an object from object storage - The object was not found
PersistentChecksumErrorInDownloadedObject	MAJOR	Checksum error detected by COMPUTE node in a downloaded OBS data block

Org

Type	Severity	Description
OrgCreated	INFO	Org Created .
OrgDeleted	INFO	Org Deleted.
OrgRenamed	INFO	Org Renamed.
OrgSsdQuotaChanged	INFO	Org SSD Quota Changed.
OrgTotalQuotaChanged	INFO	Org Total Quota Changed.

RAID

Type	Severity	Description
BitmapChecksumMismatch	MAJOR	Bitmap checksum mismatch detected.
DataGenerationNumberBug	WARNING	Bug in the advancement of the applied data generation number report from a bucket.

Type	Severity	Description
DataProtectionLevelDecreased	MINOR	Data protection level decreased.
DataProtectionLevelIncreased	INFO	Data protection level increased.
DisksFailureDetected	MINOR	Disk(s) failures detected.
DisksRecoveryDetected	INFO	Disk(s) quick recovery detected.
EnoughActiveFailureDomains	MINOR	Enough active failure domains.
FixedFalseFreeBlock	CRITICAL	Found and fixed a false free block.
HotSpareFailureDomainsUpdated	INFO	Hot spare failure domains updated.
NoDataProtection	CRITICAL	No data protection.
QuorumGenerationNumberBug	WARNING	Bug in the advancement of the applied quorum generation number report from a bucket.
RaidScrubbingRateUpdated	INFO	RAID scrubber limit updated.
RaidStarted	INFO	RAID started on bucket.
SwitchPlacementHanging	MINOR	SwitchPlacement has no non-dirty chunks.
SwitchPlacementRetrying	MINOR	SwitchPlacement retrying.
TooFewActiveFailureDomains	CRITICAL	Too few active failure domains.
TooManyFailures	CRITICAL	Too many failures, some data is unavailable.
UsedSSDCapacityCriticalOverflow	CRITICAL	SSD capacity use is critically overflowing, internal spares are running out. Cluster may soon become unavailable for writes.
UsedSSDCapacityNoLongerOverflows	INFO	SSD capacity use is no longer overflowing

Type	Severity	Description
UsedSSDCapacityOverflow	major	SSD capacity use is overflowing, internal capacity spares are being utilized

Resources

Type	Severity	Description
APIServerStartFailed	WARNING	Failed to start the API server
APIServerStarted	INFO	Successfully started the API server
BandwidthSelected	INFO	Bandwidth set for host.
CoreAllocated	INFO	Allocated core.
DisabledNumaBalancing	INFO	Disabled NUMA Balancing.
DriverLoaded	INFO	Driver loaded.
FailedToLoadDriver	WARNING	Failed to load the wekafs driver.
HugepagesAllocated	INFO	Hugepages allocated.
HugepagesAllocationRetries	WARNING	Hugepages allocation retried.
HugepagesAllocationStarted	INFO	Hugepages allocation started.
InactiveHostCannotJoinCluster	INFO	Inactive host cannot join the cluster.
LoadingStableResourcesFailed	INFO	Failed loading stable resources.
NetworkDeviceAllocated	INFO	Allocated network device.
NetworkDeviceNotUsedByAnySlots	MINOR	Network device not used by any slots.
NoIPsConfiguredForHostJoinWithNoDefaultNet	WARNING	No IP configured for node {nid} with no default-net
RevertToStableResources	INFO	Reverted to stable resources.

Type	Severity	Description
UnlimitedBandwidthSelected	INFO	Bandwidth set to unlimited.

Security

Type	Severity	Description
CaCertSet	INFO	CA cert was added to the cluster
CaCertUnset	INFO	CA cert was unset
TLSSet	INFO	TLS was set
TLSUnset	INFO	TLS was unset

SMB

Type	Severity	Description
SmbAdJoined	INFO	Active Directory configuration change.
SmbAdLeft	INFO	Active Directory configuration change.
SmbClusterConfigured	INFO	SMB cluster configuration change.
SmbClusterCreated	INFO	SMB cluster configuration change.
SmbClusterDestroyed	INFO	SMB cluster configuration change.
SmbConfigGenerationUpdated	INFO	SMB Config configuration change
SmbShareAdded	INFO	Share configuration change.
SmbShareConfigured	INFO	Share configuration change.
SmbShareHostnameACERemovedRemoved	INFO	SambaHostnameACE configuration change
SmbShareHostnameACEResetDestroyed	INFO	SambaHostnameACE configuration change

Type	Severity	Description
SmbShareRemoved	INFO	Share configuration change.
SmbTrustedDomainAdded	INFO	TrustedDomain configuration change.
SmbTrustedDomainRemoved	INFO	TrustedDomain configuration change.

Statistics

Type	Severity	Description
StatLimitExceeded	WARNING	A set limit on a stat was exceeded.

System

Type	Severity	Description
BlockTaskAborted	INFO	A bucket task aborted successfully.
BlockTaskComplete	INFO	A bucket task completed successfully.
BucketsCreated	INFO	System has created buckets.
ClusterTaskAborted	INFO	Cluster task aborted
ClusterTaskPaused	INFO	Cluster task paused
ClusterTaskResumed	INFO	Cluster task resumed
ClusterTasksCpuLimitUpdated	INFO	Cluster tasks CPU limit set
ClusterwideTaskChanged	INFO	Clusterwide task changed.
HaveEnoughSSDCapacity	MINOR	Enough SSD capacity now exists for all provisioned file systems.
IOStarted	INFO	System has started.
IOStopped	INFO	System has stopped.

Type	Severity	Description
NotEnoughSSDCapacity	CRITICAL	Not enough SSD capacity exists for all provisioned file systems.
QOSConfigReset	INFO	QoS configuration reset
QOSConfigSet	INFO	QoS configuration set
StartIORequested	INFO	The user has requested that IO be started.
StopIORequested	INFO	The user has requested that IO be stopped.
SystemInfoReport	INFO	Management node started; reporting OS info.

Traces

Type	Severity	Description
TracesConfigurationActivated	INFO	Traces configuration change
TracesConfigurationDeactivated	INFO	Traces configuration change
TracesConfigurationReset	INFO	Traces configuration change
TracesConfigurationUpdated	INFO	Traces configuration change
TracesFreezePeriodReset	INFO	Traces freeze period has been reset
TracesFreezePeriodSet	INFO	Traces freeze period has been set

Upgrade

Type	Severity	Description
ClientUpgradeRequested	INFO	Client upgrade requested
ExternalUpgradeCancelled	INFO	External Upgrade was cancelled.

Type	Severity	Description
ExternalUpgradeFinished	INFO	External Upgrade complete.
ExternalUpgradeStarting	INFO	External Upgrade was started.
FinishedExternalHostUpgrade	INFO	External host upgrade complete.
StartingExternalHostUpgrade	INFO	External host upgrade started.
WekaVersionDowngraded	WARNING	Weka is now running a lower version

User

Type	Severity	Description
LDAPAuthDisabled	INFO	LDAP authentication disabled.
LDAPAuthEnabled	INFO	LDAP authentication enabled.
LDAPConfigUpdated	INFO	LDAP configuration updated.
UserCreated	INFO	User Created.
UserDeleted	INFO	User Deleted.
UserLoggedIn	INFO	User logged in.
UserLoginFailed	INFO	User login failed.
UserLoginLocked	MINOR	User login locked
UserPasswordChangedByAnotherUser	INFO	User password changed by an admin.
UserPasswordChanged	INFO	User changed password.
UserRoleChanged	INFO	User role changed.

Chapter 15: Statistics

The statistics available in the Content Software for File system and how to work with them is described.

Overview

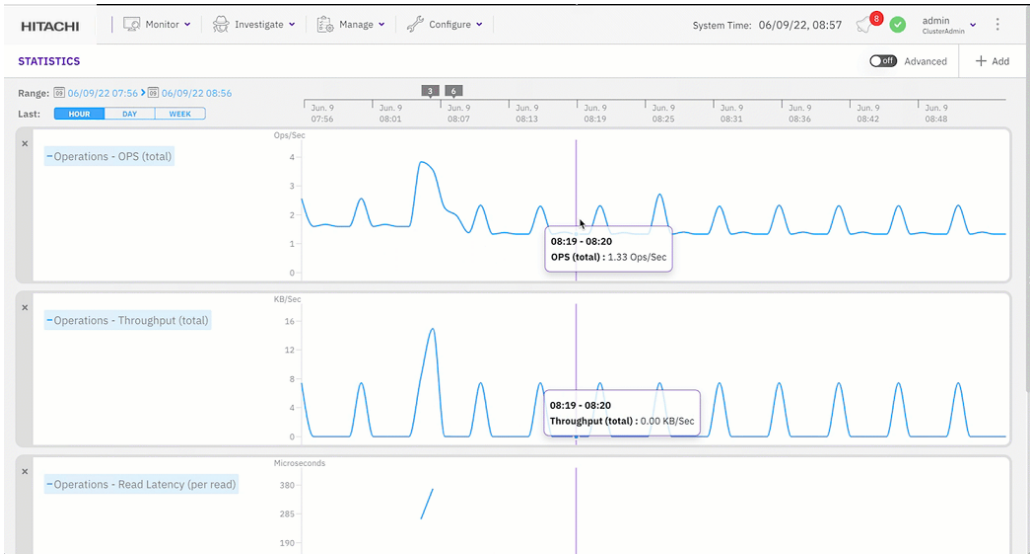
As the Content Software for File system runs, it collects hundreds of statistics on system performance. These statistics help analyze the Content Software for File system performance and determine the source of any issue.

Five different categories of statistics are available for review:

- CPU
- Object Store
- Operations
- Operations (Driver)
- Operations (NFS)
- Operations (NFSw)
- SSD

When you select each category, a list of the possible statistics related to the category is displayed, from which you can select a specific chart.

The default statistics page displays charts of the last hour of operation, presenting the system operation average value per second in one minute range.



This **Statistics** view screen offers a number of options to drill-down into the statistics, according to category. Options include:

- Mousing over the scrollable graph area to view various performance metrics of the Content Software for File cluster.
- Troubleshooting or obtaining a correlation between events and performance (using the top line which provides links to events that occurred).
- Adding more statistics to the view (using the Statistics menu).
- Displaying different statistics simultaneously and toggling between them. By default, the graph area shows Ops/sec for the last hour. Using the Hour, Day, Week buttons at the bottom-right enables changing of the time interval.
- Displaying, hiding. deleting, and zooming-in on statistics from defined timelines and dates.
- Bookmarking specific statistics for future reference and sharing with others (using the URL).

Drill-down options

This Statistics page provides several options to drill down into the charts according to the selected category.

The options include:

- Move the mouse over the scrollable chart area to view the performance metrics of the Weka cluster.
- Troubleshoot or obtain a correlation between events and performance using links to events that occurred.
- Add charts to the Statistics page, or remove charts.
- Display different charts up to five on the statistics page. The default statistics page shows OPS (total), Throughput (total), and read/write latency for the last hour. You can change the interval by selecting the Hour, Day, or Week buttons or specify a timeframe.
- Display and zoom in on statistics from defined timelines and dates.
- Bookmark specific statistics for future reference and share with others (using the URL).

Note: The page shows only the statistics of the backend and clients that are part of the cluster. The page does not show statistics in the following cases:

- A host is removed.
- A client is not connected to the cluster for more than the `retention period`.

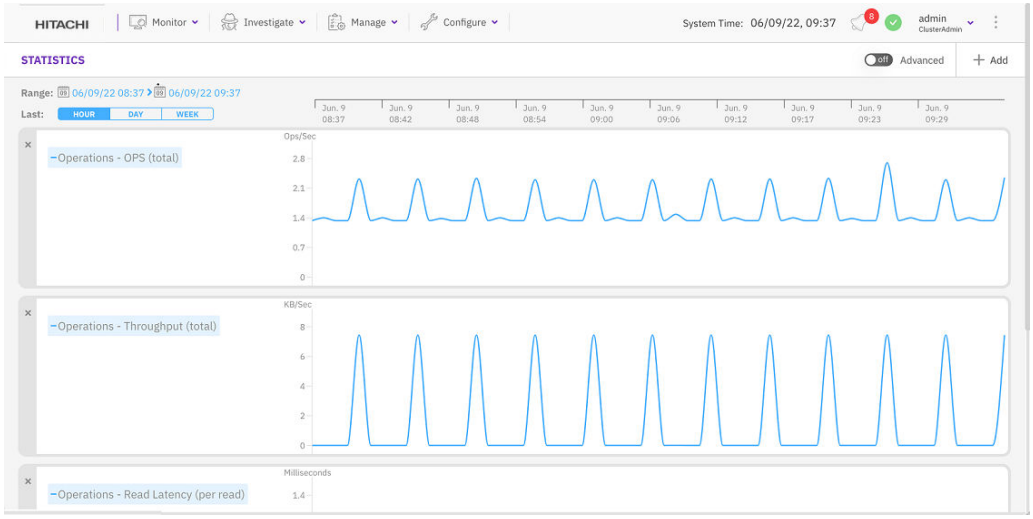
The Weka cluster does not hold historical statistics data.

Working with statistics using the GUI

How to manage the statistics using the GUI.

Viewing statistics

From the menu, select Investigate > Statistics.

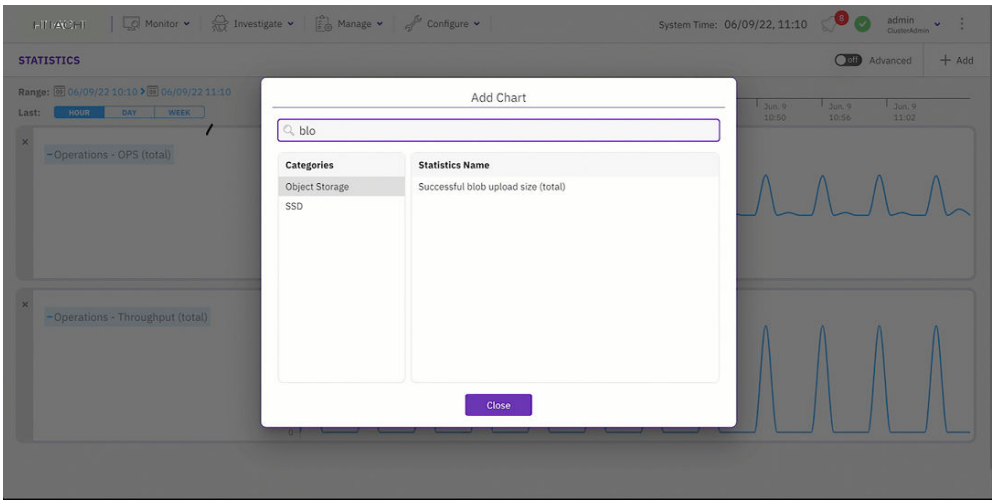


Adding a chart to the statistics page

You can add charts to the statistics page to display up to a maximum of five charts.

Procedure

1. From the menu, select **Investigate > Statistics**.
2. On the Statistics page, select **+Add**.
3. In the Add Chart dialog, do one of the following:
   - From the Categories pane, select a category, and then from the Statistics Name pane select the required chart.
   - Search for a chart using the Filter. Type a keyword or two related to the chart, and then from the Statistics Name pane select the required chart.

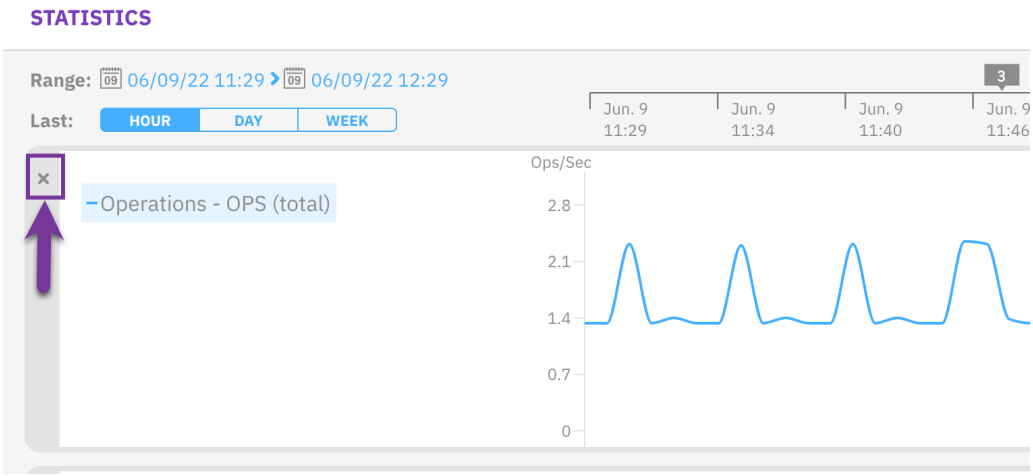


Removing a chart from the statistics page

You can remove a chart that is no longer required to free space for adding another chart to the statistics page. For example, if the Statistics page already has the maximum number of five charts.

Procedure

1. On the upper left corner of the chart, select **X**.

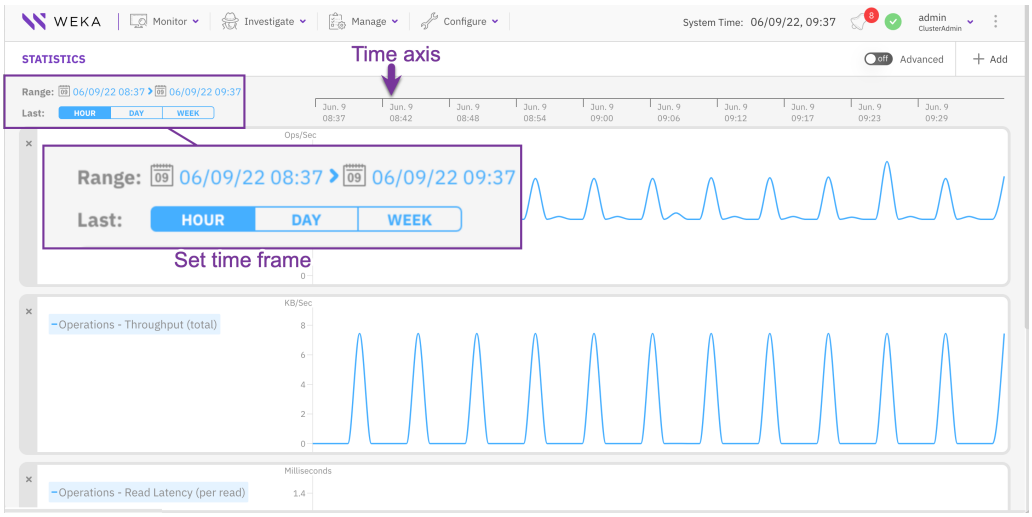


Setting the timeframe

The Statistics page contains a time axis for all the displayed charts. To investigate charts in a specific timeframe, you can set the interval in the time axis to the last hour, last day, or last week. You can also set a timeframe for a specific period (start and end time).

Procedure

1. To display the charts for the last period: **Hour**, **Day**, or **Week**, in the **Last** line, select the relevant button.
2. To display the charts for a specific period, in the **Range** line select the calendar, and set the start time and end time for the timeframe.

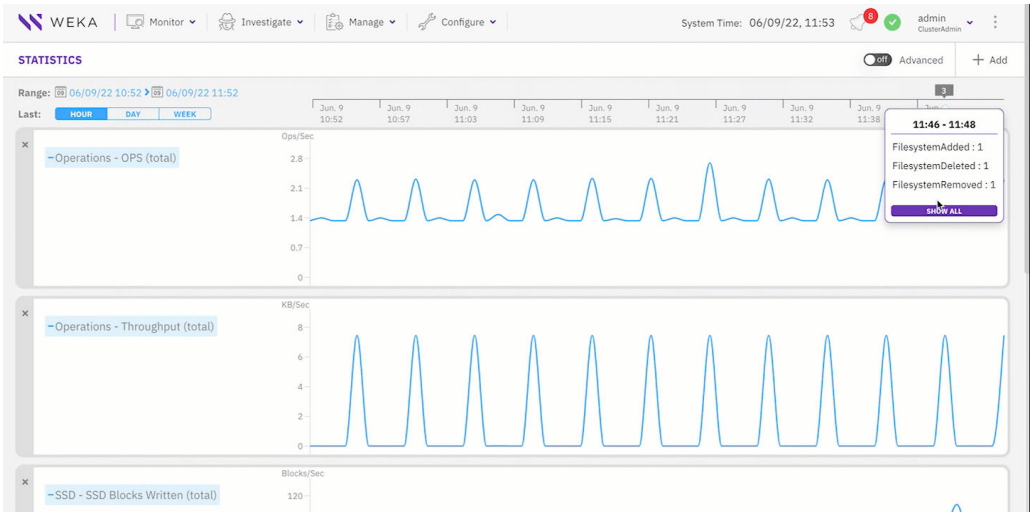


Displaying events from a chart

If events occur during the period of the displayed charts, a purple box indicates the number of the events. To investigate the events, show and correlate them with the statistics data.

Procedure

1. On the time axis, select the purple box (it only appears if events occur).
2. From the popup box, select **Show All**.



List of statistics

Attribute cache

Type	Description	Units
GP_GETATTR_CACHE_MISS	Number of general purpose getAttr cache misses per second	Ops/Sec
GP_GETATTR	Number of general purpose getAttr calls per second	Ops/Sec

Block cache

Type	Description	Units
BUCKET_CACHED_METADATA_BLOCKS	Bucket number of cached metadata blocks	Blocks
BUCKET_CACHED_REGISTRY_L2_BLOCKS	Bucket number of cached registry L2 blocks	Blocks
BUCKET_CACHE_METADATA_HITS	Bucket block cache metadata hits	Queries
BUCKET_CACHE_METADATA_MISSES	Bucket block cache metadata misses	Queries
BUCKET_CACHE_REGISTRY_L2_HITS	Bucket block cache registry L2 hits	Queries
BUCKET_CACHE_REGISTRY_L2_MISSES	Bucket block cache registry L2 misses	Queries
BUCKET_REGISTRY_L2_BLOCKS_NUM	Bucket number of registry L2 blocks	Blocks

Block writes

	Description	
BLOCK_FULL_WRITES	Full block writes	Writes
BLOCK_PARTIAL_WRITES	Partial block writes	Writes

Bucket

Type	Description	Units
BUDGET_UNDERFLOW_BLOCKS		Blocks/Sec
CHOKING_LEVEL_ALL	Throttling level applied on all types of IOs	%
CHOKING_LEVEL_NON_MUTATING	Throttling level applied on non-mutating only types of IOs	%
DESTAGED_BLOCKS_COUNT		Blocks/Sec
DESTAGE_COUNT		Destages/Sec
DIR_MOVE_TIME		Ops
EXTENT_BLOCKS_COUNT		Blocks
FREEABLE_LRU_BUFFERS		Buffers
HASH_BLOCKS_COUNT		Blocks
INODE_BLOCKS_COUNT		Blocks
INODE_REFRESHER_QUEUE_LENGTH		Items
JOURNAL_BLOCKS_COUNT		Blocks
JOURNAL_ITERATIONS	Histogram of number of batches of stripes committed in a single request	
READS	Number of read operations per second	Ops/Sec
READ_BYTES	Number of bytes read per second	Bytes/Sec
READ_LATENCY	Average latency of READ operations	Microseconds
REGISTRY_L1_BLOCKS_COUNT		Blocks
REGISTRY_L2_BLOCKS_COUNT		Blocks

Type	Description	Units
REGISTRY_SEARCHES_COUNT		Queries/Sec
RESIDENT_BLOCKS_COUNT		Blocks/Sec
SNAPSHOT_CREATION_TIME		Snaps
SPATIAL_SQUELCH_BLOCKS_COUNT		Blocks
SUCCESSFUL_DATA_WEDGINGS		Attempts/Sec
SUPERBLOCK_BLOCKS_COUNT		Blocks
TAKEOVERS_SUCCESSFUL		Takeover Attempts/Sec
TAKEOVER_ATTEMPTS		Takeover Attempts/Sec
TEMPORAL_SQUELCH_BLOCKS_COUNT		Blocks
UNSUCCESSFUL_DATA_WEDGINGS		Attempts/Sec
USER_DATA_BUFFERS_IN_USE		Buffers
WRITES	Number of write operations per second	Ops/Sec
WRITE_BYTES	Number of byte writes per second	Bytes/Sec
WRITE_LATENCY	Average latency of WRITE operations	Microseconds

Bucket failovers

Type	Description	Units
BUCKET_FAILOVERS	Amount of times swapping from a remote primary node to a secondary	Failovers

Type	Description	Units
INVALID_BUCKET_TERM	Number of times a remote bucket rejected a request becaue the term was invalid	Exceptions
REMOTE_BUCKET_IS_SECONDARY	Number of times a remote bucket reported it is secondary and cannot serve us	Exceptions

Bucket rebalances

Type	Description	Units
BUCKET_INITS	Number of bucket initializations	Times
BUCKET_INIT_LATENCY_HIST		Milliseconds
BUCKET_INIT_LATENCY	Average latency of bucket initialization	Seconds

CPU

Type	Description	Units
CPU_UTILIZATION	Percentage of the CPU time utilized for handling I/Os	%

Chocking

Type	Description	Units
CHOKING_LEVEL_ALL	Throttling level applied on all types of IOs, both mutating and non-mutating	
CHOKING_LEVEL_NON_MUTATING	Throttling level applied on non-mutating only types of IOs	

Clients

Type	Description	Units
CLIENTS_CONNECTED	Clients connected	Clients/Sec
CLIENTS_DISCONNECTED	Clients left or were removed	Clients/Sec
CLIENTS_LEFT	Clients left	Clients/Sec
CLIENTS_RECONNECTED	Clients reconnected instead of an old instance of theirs	Clients/Sec
CLIENTS_REMOVED	Clients removed	Clients/Sec

Config

Type	Description	Units
AVERAGE_CHANGES_IN_CHANGESET	Average changes in changeset	Changes/Sec
AVERAGE_CHANGES_IN_GENERATION	Average changes in generation	Changes/Sec
BACKEND_NODE_REJOIN_TIME		Milliseconds
CHANGESET_COMMIT_LATENCY	Average latency of committing a config changeset	Microseconds
CLIENT_NODE_REJOIN_TIME		Milliseconds
GENERATION_COMMIT_LATENCY	Average latency of committing a config generation	Microseconds
HEARTBEAT_PROCESSING_TIME_OLD		Seconds
HEARTBEAT_PROCESSING_TIME		Seconds
LEADER_HEARTBEAT_PROCESSING_TIME_OLD		Seconds
LEADER_HEARTBEAT_PROCESSING_TIME		Seconds

Type	Description	Units
TOTAL_CHANGESETS_COMMITTED	Total number of changesets committed	Change Sets
TOTAL_COMMITTED_CHANGES	Total number of config changes committed	Changes
TOTAL_GENERATIONS_COMMITTED	Number of generations committed per second	Generations

Filesystem OBS

Type	Description	Units
BACKPRESSURED_BUCKETS_IN_FSS	Number of backpressured buckets	Buckets
CONCURRENT_DEMOTES	How many demotes are executed concurrently	Demotes
DEMOTE_EXTENT_OBS_FETCH_BACKPRESSURE	Number of extent BACKPRESSURE fetch operations per second	Ops/Sec
DEMOTE_EXTENT_OBS_FETCH_IMMEDIATE_RELEASE	Number of extent IMMEDIATE_RELEASE fetch operations per second	Ops/Sec
DEMOTE_EXTENT_OBS_FETCH_MANHOLE	Number of extent MANHOLE fetch operations per second	Ops/Sec
DEMOTE_EXTENT_OBS_FETCH_MIGRATE	Number of extent MIGRATE fetch operations per second	Ops/Sec
DEMOTE_EXTENT_OBS_FETCH_POLICY	Number of extent POLICY fetch operations per second	Ops/Sec
DEMOTE_EXTENT_OBS_FETCH_RECLAMATION_REUPLOAD	Number of extent RECLAMATION_REUPLOAD fetch operations per second	Ops/Sec
DEMOTE_EXTENT_OBS_FETCH_STOW	Number of extent STOW fetch operations per second	Ops/Sec
DEMOTE_EXTENT_OBS_FETCH	Number of extent fetch operations per second	Ops/Sec

Type	Description	Units
DEMOTE_WAITING_FOR_SLOT	Average time waiting for a demotion concurrency slot	Microseconds
DESERIALIZED_EXTENTS_WITH_INVALID_BLOBS	Number of deserialized extents with invalid blob id	Extents
DOWNLOADS	Number of promotes operations per second	Ops/Sec
DOWNLOAD_LATENCY	Latency of promote operations	Microseconds
FAILED_DOWNLOADS	Number of failed promotes operations per second	Ops/Sec
FAILED_UPLOADS	Number of failed demotes operations per second	Ops/Sec
OBS_4K_IOPS_READ	Number of object storage dedicated 4K read operations per second	Ops/Sec
OBS_BACKPRESSURE_FREED	Number of bytes freed from disk due to backpressure	Bytes/Sec
OBS_BLOB_HEADER_DOWNLOAD_LATENCY	Average latency of blob header download	Microseconds
OBS_BLOB_SCAVENGE_LATENCY	Average latency of blob scavenges	Microseconds
OBS_BLOB_TIERING_DURATION		Microseconds
OBS_COMPLETELY_ALIVE_BLOBS	Percentage of blobs with only live extents linked to them	%
OBS_COMPLETELY_DEAD_BLOBS	Percentage of blobs with no live extent linked to them	%
OBS_EXTENTS_PREFETCH	Number of pre-fetched extents	Ops/Sec
OBS_FREED	Number of bytes freed from disk because they are in the OBS	Bytes/Sec
OBS_IMMEDIATE_RELEASE_FREED	Number of bytes freed from disk due to immediate release	Bytes/Sec

Type	Description	Units
OBS_INODES_PREFETCH	Number of pre-fetched inodes	Ops/Sec
OBS_INODES_RELEASE	Number of pre-fetched inodes	Ops/Sec
OBS_ONGOING_RECLAMATIONS	Number of ongoing reclamations	Ops
OBS_POLICY_FREED	Number of bytes freed from disk due to policy	Bytes/Sec
OBS_PROMOTE_EXTENT_WRITE_LATENCY		Microseconds
OBS_PROMOTE_EXTENT_WRITE		Ops/Sec
OBS_PROMOTE_WRITE		Bytes/Sec
OBS_READ	Reads that needed data from the OBS	Ops/Sec
OBS_RECLAMATION_PURGED_BYTES	Number of bytes purged per second	Bytes/Sec
OBS_RECLAMATION_SCAVENGED_BLOBS	Number of blobs scavenged per second	Ops/Sec
OBS_RECLAMATION_SCAVENGED_BYTES	Number of blobs scavenged per second	Bytes/Sec
OBS_RECLAMATION_WAIT_FOR_DESTAGE	Average time waiting for destage on space reclamation	Microseconds
OBS_RELOC_DOWNLOAD	Number of relocation blobs downloaded per second	Ops/Sec
OBS_RELOC_UPLOAD	Number of relocation blobs uploaded per second	Ops/Sec
OBS_SCAVENGED_BLOB_WASTE_LEVEL	Histogram of waste level found in blobs	
OBS_SHARED_DOWNLOADS_LATENCY		Microseconds
OBS_SHARED_DOWNLOADS		Ops/Sec

Type	Description	Units
OBS_TRUNCATE	Truncates that needed data from the OBS	Ops/Sec
OBS_UNEXPECTED_TAG_ON_DOWNLOAD	Unexpected tag when downloading an extent	Occurences
OBS_WRITE	Writes that needed data from the OBS	Ops/Sec
STOW_SERIALIZED_EXTENT_DATA	Number of extent descriptors uploaded that contain data	Extent descriptors
STOW_SERIALIZED_EXTENT_DESCS	Number of extent descriptors uploaded	Extent descriptors
STOW_SERIALIZED_EXTENT_REDIRECTS	Number of extent descriptors uploaded that redirect to previous snapshot	Extent descriptors
TIERED_FS_BREAKING_POLICY	Tiered Filesystem Breaking Policy Counter	Activations
TIMEOUT_DOWNLOADS	Number of timeout'ed promotes operations per second	Ops/Sec
TIMEOUT_OPERATIONS	Total timeouted operations per second	Ops/Sec
TIMEOUT_UPLOADS	Number of timeout'ed demotes operations per second	Ops/Sec
UNEXPECTED_BLOCK_VERSION_POST_UPGRADE	Unexpected block version after upgrade completed	Occurences
UNEXPECTED_HASHBLOCK_KV_VERSION_POST_UPGRADE	Unexpected hash block KV version after upgrade completed	Occurences
UPLOADS	Number of upload attempts per second	Ops/Sec
UPLOAD_CHOKING_LATENCY	Average latency of waiting for demote choking budget	Microseconds
UPLOAD_LATENCY	Latency of demote	Microseconds

Frontend

Type	Description	Units
FE_IDLE_CYCLES		Cycles/Sec
FE_IDLE_TIME	Percentage of the CPU time not utilized for handling I/Os on frontend	%

Frontend encryption

Type	Description	Units
FE_BLOCKS_DECRYPTED	Number of blocks decrypted in the frontend	Blocks
FE_BLOCKS_ENCRYPTED	Number of blocks encrypted in the frontend	Blocks
FE_BLOCK_CRYPTO_LATENCY	Average latency of frontend block crypto	Microseconds
FE_BLOCK_DECRYPT_DURATION	Duration of decryption of blocks in the frontend	Microseconds
FE_BLOCK_ENCRYPT_DURATION	Duration of encryption of blocks in the frontend	Microseconds
FE_FILENAMES_DECRYPTED	Number of filenames decrypted in the frontend	Filenames
FE_FILENAMES_ENCRYPTED	Number of filenames encrypted in the frontend	Filenames
FE_FILENAME_CRYPTO_LATENCY	Average latency of frontend filename crypto	Microseconds
FE_FILENAME_DECRYPT_DURATION	Duration of decryption of filenames in the frontend	Microseconds
FE_FILENAME_ENCRYPT_DURATION	Duration of encryption of filenames in the frontend	Microseconds

Garbage collection

Type	Description	Units
GC_FREE_SIZE_AFTER_SCAN	GC pool size after the scan ends	Bytes
GC_FREE_SIZE_BEFORE_SCAN	GC pool size before the scan starts	Bytes
GC_SCAN_TIME	GC scan time	Msec
GC_USED_SIZE_AFTER_SCAN	GC used size after the scan ends	Bytes
GC_USED_SIZE_BEFORE_SCAN	GC used size before the scan starts	Bytes

JRPC

Type	Descriptions	Units
JRPC_SERVER_PROCESSING_AVG		Microseconds
JRPC_SERVER_PROCESSING_TIME		

Journal

Type	Description	Units
JOURNAL_CURRENT_OPS	Operations currently in journal	Journal Entries
JOURNAL_OPS_IN	Operations added to the journal	Journal Entries/Sec
JOURNAL_OPS_OUT	Operations removed from the journal	Journal Entries/Sec

Memory

Type	Description	Units
RSS_CURRENT		MB
RSS_PEAK		MB

Network

Type	Description	Units
BAD_RECV_CSUM	Number of packets received with a bad checksum	Packets/Sec
CORRUPT_PACKETS	Number of packets received and deemed corrupted	Packets/Sec
DOUBLY_RECEIVED_PACKETS	Number of packets that were received multiple times	Packets/Sec
DROPPED_LARGE_PACKETS	Number of large packets dropped in the socket backend	Packets/Sec
DROPPED_PACKETS	Number of packets received that we dropped	Packets/Sec
ECN_ENCOUNTERED	Number of ECN Encountered packets	Packets/Sec
FAULT_RECV_DELAYED_PACKETS	Number of received packets delayed due to a fault injection	Packets/Sec
FAULT_RECV_DROPPED_PACKETS	Number of received packets dropped due to a fault injection	Packets/Sec
FAULT_SENT_DELAYED_PACKETS	Number of sent packets delayed due to a fault injection	Packets/Sec
FAULT_SENT_DROPPED_PACKETS	Number of sent packets dropped due to a fault injection	Packets/Sec
GOODPUT_RX_RATIO	Percentage of goodput RX packets out of total data packets received	%

Type	Description	Units
GOODPUT_TX_RATIO	Percentage of goodput TX packets out of total data packets sent	%
GW_MAC_RESOLVE_FAILURES	Number of times we failed to ARP resolve the gateway IP	Failures
GW_MAC_RESOLVE_SUCCESSES	Number of times we succeeded to ARP resolve the gateway IP	Successes
NODE_RECONNECTED	Number of reconnections	Reconnects/Sec
PACKETS_PUMPED	Number of packets received in each call to recv	Packets
PEER_RTT	RTT per peer node	Microseconds
PORT_RX_BYTES	Number of bytes received	Bytes/Sec
PORT_RX_PACKETS	Number of packets received	Packets/Sec
PORT_TX_BYTES	Number of bytes transmitted	Bytes/Sec
PORT_TX_PACKETS	Number of packets transmitted	Packets/Sec
PUMPS_TXQ_FULL	Number of times we couldn’t send any new packets to the NIC queue	Pumps/Sec
PUMPS_TXQ_PARTIAL	Number of times we only sent some of our queued packets to the NIC queue	Pumps/Sec
PUMP_DURATION	Duration of each pump	
PUMP_INTERVAL	Interval between pumps	
RDMA_ADD_CHUNK_FAILURES		Failures/Sec
RDMA_BINDING_FAILOVERS		Fail-overs/Sec
RDMA_CANCELED_COMPLETIONS		Completions/Sec
RDMA_CLIENT_BINDING_INVALIDATIONS		Invalidations/Sec
RDMA_COMPLETIONS		Completions/Se

Type	Description	Units
RDMA_COMP_DURATION		
RDMA_COMP_FAILURES		Failures/Sec
RDMA_COMP_LATENCY	Average time of RDMA requests completion	Microseconds
RDMA_NET_ERR_RETRY_EXCEEDED		Occurences/Sec
RDMA_POOL_ALLOC_FAILED		Failures/Sec
RDMA_POOL_LOW_CAPACITY		Failures/Sec
RDMA_PORT_WAITING_FIBERS		Waiting fibers
RDMA_REQUESTS		Requests/Sec
RDMA_RX_BYTES		Bytes/Sec
RDMA_SERVER_BINDING_RESTARTS		Restarts/Sec
RDMA_SUBMIT_FAILURES		Failures/Sec
RDMA_SUBMIT_TIMEOUTS		Timeouts/Sec
RDMA_TX_BYTES		Bytes/Sec
RECEIVED_CONTROL_PACKETS	Number of received control packets	Packets/Sec
RECEIVED_DATA_PACKETS	Number of received data packets	Packets/Sec
RECEIVED_PACKETS	Number of packets received	Packets/Sec
RECEIVED_PACKET_GENERATIONS	The generation ("resend count") of the first incarnation of the packet seen by the receiver (indicates packet loss)	
REORDERED_PACKETS	Number of reordered packets	Packets/Sec
RESEND_BATCH_SIZE	Number of packets sent in a resend batch	

Type	Description	Units
RESENT_DATA_PACKETS	Number of data packets resent	Packets/Sec
SEND_QUEUE_TIMEOUTS	Number of packets cancelled due to envelope timeout and were not in the send window	Packets/Sec
SEND_WINDOW_TIMEOUTS	Number of packets cancelled due to envelope timeout while in the send window	Packets/Sec
SENT_ACKS	Number of ACK packets sent	Packets/Sec
SENT_CONTROL_PACKETS	Number of control packets sent	Packets/Sec
SENT_DATA_PACKETS	Number of data packets sent	Packets/Sec
SENT_PACKETS	Number of sent packets	Packets/Sec
SENT_REJECTS	Number of rejects sent	Packets/Sec
SHORT_CIRCUIT_SENDS	Number of packets sent to the same node	Packets/Sec
SLOW_PATH_CSUM	Number of packets that went through checksum calculation on the CPU	Packets/Sec
TIMELY_RESENDS	Number of packets resent due to timely resend	Packets/Sec
TIME_TO_ACK	Histogram of time to ack a data packet	Packets/Sec
TIME_TO_FIRST_SEND	Time from queueing to first send	
UCX_SEND_CB		Packets/Sec
UCX_SEND_ERROR		Packets/Sec
UCX_SENT_PACKETS_ASYNC		Packets/Sec
UCX_SENT_PACKETS_IMMEDIATE		Packets/Sec
UCX_TXQ_FULL		Packets/Sec

Type	Description	Units
UDP_SENDMSG_FAILED_EAGAIN	Number of packets that failed to be sent on the socket backend with EAGAIN	Packets/Sec
UDP_SENDMSG_FAILED_OTHER	Number of packets that failed to be sent on the socket backend with an unknown error	Packets/Sec
UDP_SENDMSG_PARTIAL_SEND	Number of packets that we failed to send but in the same pump some packets were sent	Packets/Sec
UNACKED_RESENDS	Number of packets resent after receiving an ack	Packets/Sec
ZERO_CSUM	Number of checksum zero received	Packets/Sec

Object storage

Type	Description	Units
FAILED_OBJECT_DELETES	Number of failed object deletes per second (any failure reason)	Ops/Sec
FAILED_OBJECT_DOWNLOADS	Number of failed object download per second (any failure reason)	Ops/Sec
FAILED_OBJECT_HEAD_QUERIES	Number of failed object head queries per second (any failure reason)	Ops/Sec
FAILED_OBJECT_OPERATIONS	Total failed operations per second	Ops/Sec
FAILED_OBJECT_UPLOADS	Number of failed object uploads per second (any failure reason)	
OBJECT_DELETES	Number of object deletes per second	Ops/Sec

Type	Description	Units
OBJECT_DELETE_DURATION		Milliseconds
OBJECT_DELETE_LATENCY	Latency of deleting an object	Microseconds
OBJECT_DOWNLOADS_BACKGROUND	Number of BACKGROUND objects downloaded per second	Ops/Sec
OBJECT_DOWNLOADS_FOREGROUND	Number of FOREGROUND objects downloaded per second	Ops/Sec
OBJECT_DOWNLOADS	Number of objects downloaded per second	Ops/Sec
OBJECT_DOWNLOAD_BYTES_BACKGROUND	Number of BACKGROUND bytes sent to the object storage	Bytes/Sec
OBJECT_DOWNLOAD_BYTES_FOREGROUND	Number of FOREGROUND bytes sent to the object storage	Bytes/Sec
OBJECT_DOWNLOAD_DURATION		Milliseconds
OBJECT_DOWNLOAD_LATENCY	Latency of downloading an object	Microseconds
OBJECT_DOWNLOAD_SIZE		Bytes
OBJECT_HEAD_DURATION		Milliseconds
OBJECT_HEAD_LATENCY	Latency of deleting an object	Milliseconds
OBJECT_HEAD_QUERIES	Number of object head queries per second	Ops/Sec
OBJECT_OPERATIONS	Total operations per second	Ops/Sec
OBJECT_UPLOADS_BACKPRESSURE	Number of BACKPRESSURE upload attempts per second	Ops/Sec
OBJECT_UPLOADS_IMMEDIATE_RELEASE	Number of IMMEDIATE_RELEASE upload attempts per second	Ops/Sec

Type	Description	Units
OBJECT_UPLOADS_MANHOLE	Number of MANHOLE upload attempts per second	Ops/Sec
OBJECT_UPLOADS_MIGRATE	Number of MIGRATE upload attempts per second	Ops/Sec
OBJECT_UPLOADS_POLICY	Number of POLICY upload attempts per second	Ops/Sec
OBJECT_UPLOADS_RECLAMATION_REUPLOAD	Number of RECLAMATION_REUPLOAD upload attempts per second	Ops/Sec
OBJECT_UPLOADS_STOW	Number of STOW upload attempts per second	Ops/Sec
OBJECT_UPLOADS	Number of object uploads per second	Ops/Sec
OBJECT_UPLOAD_BYTES_BACKPRESSURE	Number of BACKPRESSURE bytes sent to the object storage	Bytes/Sec
OBJECT_UPLOAD_BYTES_IMMEDIATE_RELEASE	Number of IMMEDIATE_RELEASE bytes sent to the object storage	Bytes/Sec
OBJECT_UPLOAD_BYTES_MANHOLE	Number of MANHOLE bytes sent to object storage	Bytes/Sec
OBJECT_UPLOAD_BYTES_MIGRATE	Number of MIGRATE bytes sent to the object storage	Bytes/Sec
OBJECT_UPLOAD_BYTES_POLICY	Number of POLICY bytes sent to the object storage	Bytes/Sec
OBJECT_UPLOAD_BYTES_RECLAMATION_REUPLOAD	Number of RECLAMATION_REUPLOAD bytes sent to the object storage	Bytes/Sec
OBJECT_UPLOAD_BYTES_STOW	Number of STOW bytes sent to the object storage	Bytes/Sec
OBJECT_UPLOAD_DURATION		Milliseconds
OBJECT_UPLOAD_LATENCY	Latency of uploading an object	Microseconds

Type	Description	Units
OBJECT_UPLOAD_SIZE		Bytes
OBS_READ_BYTES	Number of bytes read from the object storage	Bytes/Sec
OBS_WRITE_BYTES	Number of bytes sent to the object storage	Bytes/Sec
ONGOING_DOWNLOADS	Number of ongoing downloads	Ops
ONGOING_REMOVES	Number of ongoing removes	Ops
ONGOING_UPLOADS	Number of ongoing uploads	Ops
READ_BYTES	Number of bytes read from the object storage	Bytes/Sec
REQUEST_COUNT_DELETE	Number of HTTP DELETE requests per second	Requests/Sec
REQUEST_COUNT_GET	Number of HTTP GET requests per second	Requests/Sec
REQUEST_COUNT_HEAD	Number of HTTP HEAD requests per second	Requests/Sec
REQUEST_COUNT_INVALID	Number of HTTP INVALID requests per second	Requests/Sec
REQUEST_COUNT_POST	Number of HTTP POST requests per second	Requests/Sec
REQUEST_COUNT_PUT	Number of HTTP PUT requests per second	Requests/Sec
RESPONSE_COUNT_ACCEPTED	Number of HTTP ACCEPTED responses per second	Responses/Sec
RESPONSE_COUNT_BAD_GATEWAY	Number of HTTP BAD_GATEWAY responses per second	Responses/Sec
RESPONSE_COUNT_BAD_REQUEST	Number of HTTP BAD_REQUEST responses per second	Responses/Sec
RESPONSE_COUNT_CONFLICT	Number of HTTP CONFLICT responses per second	Responses/Sec

Type	Description	Units
RESPONSE_COUNT_CONTINUE	Number of HTTP CONTINUE responses per second	Responses/Sec
RESPONSE_COUNT_CREATED	Number of HTTP CREATED responses per second	Responses/Sec
RESPONSE_COUNT_EXPECTATION_FAILED	Number of HTTP EXPECTATION_FAILED responses per second	Responses/Sec
RESPONSE_COUNT_FORBIDDEN	Number of HTTP FORBIDDEN responses per second	Responses/Sec
RESPONSE_COUNT_FOUND	Number of HTTP FOUND responses per second	Responses/Sec
RESPONSE_COUNT_GATEWAY_TIMEOUT	Number of HTTP GATEWAY_TIMEOUT responses per second	Responses/Sec
RESPONSE_COUNT_GONE	Number of HTTP GONE responses per second	Responses/Sec
RESPONSE_COUNT_HTTP_VERSION_NOT_SUPPORTED	Number of HTTP HTTP_VERSION_NOT_SUPPORTED responses per second	Responses/Sec
RESPONSE_COUNT_INSUFFICIENT_STORAGE	Number of HTTP INSUFFICIENT_STORAGE responses per second	Responses/Sec
RESPONSE_COUNT_INVALID	Number of HTTP INVALID responses per second	Responses/Sec
RESPONSE_COUNT_LENGTH_REQUIRED	Number of HTTP LENGTH_REQUIRED responses per second	Responses/Sec
RESPONSE_COUNT_METHOD_NOT_ALLOWED	Number of HTTP METHOD_NOT_ALLOWED responses per second	Responses/Sec
RESPONSE_COUNT_MOVED_PERMANENTLY	Number of HTTP MOVED_PERMANENTLY responses per second	Responses/Sec

Type	Description	Units
RESPONSE_COUNT_NON_AUTH_INFO	Number of HTTP NON_AUTH_INFO responses per second	Responses/Sec
RESPONSE_COUNT_NOT_ACCEPABLE	Number of HTTP NOT_ACCEPABLE responses per second	Responses/Sec
RESPONSE_COUNT_NOT_FOUND	Number of HTTP NOT_FOUND responses per second	Responses/Sec
RESPONSE_COUNT_NOT_IMPLEMENTED	Number of HTTP NOT_IMPLEMENTED responses per second	Responses/Sec
RESPONSE_COUNT_NOT_MODIFIED	Number of HTTP NOT_MODIFIED responses per second	Responses/Sec
RESPONSE_COUNT_NO_CONTENT	Number of HTTP NO_CONTENT responses per second	Responses/Sec
RESPONSE_COUNT_OK	Number of HTTP OK responses per second	Responses/Sec
RESPONSE_COUNT_PARTIAL_CONTENT	Number of HTTP PARTIAL_CONTENT responses per second	Responses/Sec
RESPONSE_COUNT_PAYMENT_REQUIRED	Number of HTTP PAYMENT_REQUIRED responses per second	Responses/Sec
RESPONSE_COUNT_PRECONDITION_FAILED	Number of HTTP PRECONDITION_FAILED responses per second	Responses/Sec
RESPONSE_COUNT_PROXY_AUTH_REQUIRED	Number of HTTP PROXY_AUTH_REQUIRED responses per second	Responses/Sec
RESPONSE_COUNT_REDIRECT_MULTIPLE_CHOICES	Number of HTTP REDIRECT_MULTIPLE_CHOICES responses per second	Responses/Sec

Type	Description	Units
RESPONSE_COUNT_REQUESTED_RANGE_NOT_SATISFIABLE	Number of HTTP REQUESTED_RANGE_NOT_SATISFIABLE responses per second	Responses/Sec
RESPONSE_COUNT_REQUEST_HEADER_FIELDS_TOO_LARGE	Number of HTTP REQUEST_HEADER_FIELDS_TOO_LARGE responses per second	Responses/Sec
RESPONSE_COUNT_REQUEST_TIMEOUT	Number of HTTP REQUEST_TIMEOUT responses per second	Responses/Sec
RESPONSE_COUNT_REQUEST_TOO_LARGE	Number of HTTP REQUEST_TOO_LARGE responses per second	Responses/Sec
RESPONSE_COUNT_RESET_CONTENT	Number of HTTP RESET_CONTENT responses per second	Responses/Sec
RESPONSE_COUNT_SEE_OTHER	Number of HTTP SEE_OTHER responses per second	Responses/Sec
RESPONSE_COUNT_SERVER_ERROR	Number of HTTP SERVER_ERROR responses per second	Responses/Sec
RESPONSE_COUNT_SERVICE_UNAVAILABLE	Number of HTTP SERVICE_UNAVAILABLE responses per second	Responses/Sec
RESPONSE_COUNT_SWITCHING_PROTOCOL	Number of HTTP SWITCHING_PROTOCOL responses per second	Responses/Sec
RESPONSE_COUNT_TEMP_REDIRECT	Number of HTTP TEMP_REDIRECT responses per second	Responses/Sec
RESPONSE_COUNT_UNAUTHORIZED	Number of HTTP UNAUTHORIZED responses per second	Responses/Sec
RESPONSE_COUNT_UNPROCESSABLE_ENTITY	Number of HTTP UNPROCESSABLE_ENTITY responses per second	Responses/Sec

Type	Description	Units
RESPONSE_COUNT_UNSUPPORTED_MEDIA_TYPE	Number of HTTP UNSUPPORTED_MEDIA_TYPE responses per second	Responses/Sec
RESPONSE_COUNT_URI_TOO_LONG	Number of HTTP URI_TOO_LONG responses per second	Responses/Sec
RESPONSE_COUNT_USE_PROXY	Number of HTTP USE_PROXY responses per second	Responses/Sec
WAITING_FOR_BUCKET_DOWNLOAD_BANDWIDTH		Milliseconds
WAITING_FOR_BUCKET_DOWNLOAD_FLOW		Milliseconds
WAITING_FOR_BUCKET_REMOVE_FLOW		Milliseconds
WAITING_FOR_BUCKET_UPLOAD_BANDWIDTH		Milliseconds
WAITING_FOR_BUCKET_UPLOAD_FLOW		Milliseconds
WAITING_FOR_GROUP_DOWNLOAD_BANDWIDTH		Milliseconds
WAITING_FOR_GROUP_DOWNLOAD_FLOW		Milliseconds
WAITING_FOR_GROUP_REMOVE_FLOW		Milliseconds
WAITING_FOR_GROUP_UPLOAD_BANDWIDTH		Milliseconds
WAITING_FOR_GROUP_UPLOAD_FLOW		Milliseconds
WAITING_IN_BUCKET_DOWNLOAD_QUEUE		Milliseconds
WAITING_IN_BUCKET_REMOVE_QUEUE		Milliseconds
WAITING_IN_BUCKET_UPLOAD_QUEUE		Milliseconds

Type	Description	Units
WAITING_IN_GROUP_DOWNLOAD_QUEUE		Milliseconds
WAITING_IN_GROUP_REMOVE_QUEUE		Milliseconds
WAITING_IN_GROUP_UPLOAD_QUEUE		Milliseconds
WRITE_BYTES	Number of bytes sent to the object storage	Bytes/Sec

Operations(NFS)

Type	Description	Units
ACCESS_LATENCY	Average latency of ACCESS operations	Microseconds
ACCESS_OPS	Number of ACCESS operation per second	Ops/Sec
COMMIT_LATENCY	Average latency of COMMIT operations	Microseconds
COMMIT_OPS	Number of COMMIT operation per second	Ops/Sec
CREATE_LATENCY	Average latency of CREATE operations	Microseconds
CREATE_OPS	Number of CREATE operation per second	Ops/Sec
FSINFO_LATENCY	Average latency of FSINFO operations	Microseconds
FSINFO_OPS	Number of FSINFO operation per second	Ops/Sec
GETATTR_LATENCY	Average latency of GETATTR operations	Microseconds
GETATTR_OPS	Number of GETATTR operation per second	Ops/Sec
LINK_LATENCY	Average latency of LINK operations	Microseconds

Type	Description	Units
LINK_OPS	Number of LINK operation per second	Ops/Sec
LOOKUP_LATENCY	Average latency of LOOKUP operations	Microseconds
LOOKUP_OPS	Number of LOOKUP operation per second	Ops/Sec
MKDIR_LATENCY	Average latency of MKDIR operations	Microseconds
MKDIR_OPS	Number of MKDIR operation per second	Ops/Sec
MKNOD_LATENCY	Average latency of MKNOD operations	Microseconds
MKNOD_OPS	Number of MKNOD operation per second	Ops/Sec
OPS	Total number of operations	Ops/Sec
PATHCONF_LATENCY	Average latency of PATHCONF operations	Microseconds
PATHCONF_OPS	Number of PATHCONF operation per second	Ops/Sec
READDIR_LATENCY	Average latency of READDIR operations	Microseconds
READDIR_OPS	Number of READDIR operation per second	Ops/Sec
READLINK_LATENCY	Average latency of READLINK operations	Microseconds
READLINK_OPS	Number of READLINK operation per second	Ops/Sec
READS	Number of read operations per second	Ops/Sec
READ_BYTES	Number of bytes read per second	Bytes/Sec
READ_DURATION		Microseconds
READ_LATENCY	Average latency of READ operations	Microseconds

Type	Description	Units
READ_SIZES	NFS read sizes histogram	
REMOVE_LATENCY	Average latency of REMOVE operations	Microseconds
REMOVE_OPS	Number of REMOVE operation per second	Ops/Sec
RENAME_LATENCY	Average latency of RENAME operations	Microseconds
RENAME_OPS	Number of RENAME operation per second	Ops/Sec
SETATTR_LATENCY	Average latency of SETATTR operations	Microseconds
SETATTR_OPS	Number of SETATTR operation per second	Ops/Sec
STATFS_LATENCY	Average latency of STATFS operations	Microseconds
STATFS_OPS	Number of STATFS operation per second	Ops/Sec
SYMLINK_LATENCY	Average latency of SYMLINK operations	Microseconds
SYMLINK_OPS	Number of SYMLINK operation per second	Ops/Sec
THROUGHPUT	Number of byte read/writes per second	Bytes/Sec
WRITES	Number of write operations per second	Ops/Sec
WRITE_BYTES	Number of byte writes per second	Bytes/Sec
WRITE_DURATION		Microseconds
WRITE_LATENCY	Average latency of WRITE operations	Microseconds
WRITE_SIZES	NFS write sizes histogram	

Operations (NFSw)

Type	Description	Units
AVG_COPY_OPS	Average copy operations per second	Ops/Sec
AVG_DELETE_OPS	Average delete operations per second	Ops/Sec
AVG_GET_OPS	Average get operations per second	Ops/Sec
AVG_LIST_V1_OPS	Average list v1 operations per second	Ops/Sec
AVG_LIST_V2_OPS	Average list v2 operations per second	Ops/Sec
AVG_MULTIPART_UPLOAD_OPS	Average multipart upload operations per second	Ops/Sec
AVG_PUT_OBJECTPART_OPS	Average put objectpart operations per second	Ops/Sec
AVG_PUT_OPS	Average put operations per second	Ops/Sec
READ_BYTES	Number of byte reads per second	Bytes/Sec
TOTAL_BUCKET_CREATE_OPS	Total bucket create operations per second	Ops/Sec
TOTAL_BUCKET_DELETE_OPS	Total bucket delete operation per seconds	Ops/Sec
TOTAL_BUCKET_LIST_OPS	Total bucket list operations per second	Ops/Sec
TOTAL_COPY_LATENCY	Average latency of Copy operations	Microseconds
TOTAL_COPY_OPS	Total Copy operations	Ops
TOTAL_DELETE_OPS	Total delete operations	Ops
TOTAL_GET_BUCKET_ACL_OPS	Total get bucket acl operations per second	Ops/Sec
TOTAL_GET_BUCKET_NOTIFICATION_OPS	Total get bucket notifications operations per second	Ops/Sec

Type	Description	Units
TOTAL_GET_LATENCY	Average latency of Get operations	Microseconds
TOTAL_GET_OPS	Total Get operations	Ops
TOTAL_LIST_V1_OPS	Total list v1 operations	Ops
TOTAL_LIST_V2_OPS	Total list v2 operations	Ops
TOTAL_MULTIPART_UPLOAD_LATENCY	Average latency of Multipart upload operations	Microseconds
TOTAL_MULTIPART_UPLOAD_OPS	Total multipart upload operations	Ops
TOTAL_PUT_BUCKET_ACL_OPS	Total put bucket acl operations per second	Ops/Sec
TOTAL_PUT_LATENCY	Average latency of Put operations	Microseconds
TOTAL_PUT_OBJECTPART_OPS	Total put objectpart operations	Ops
TOTAL_PUT_OPS	Total put operations	Ops
WRITE_BYTES	Number of byte writes per seconds	Bytes/Sec

Operations(driver)

Type	Description	Units
DIRECT_READ_SIZES		Blocks/Sec
DIRECT_WRITE_SIZES		Blocks
DIRECT_WRITE_SIZES_RATE		Blocks/Sec
DIRECT_WRITE_SIZES		Blocks
DOORBELL_RING_COUNT		Ops
FAILED_1HOP_READS	Number of failed single hop reads per second	Ops/Sec

Type	Description	Units
FILEATOMICOPEN_LATENCY	Average latency of FILEATOMICOPEN operations	Microseconds
FILEATOMICOPEN_OPS	Number of FILEATOMICOPEN operation per second	Ops/Sec
FILECLOSE_LATENCY	Average latency of FILECLOSE operations	Microseconds
FILECLOSE_OPS	Number of FILECLOSE operation per second	Ops/Sec
FILEOPEN_LATENCY	Average latency of FILEOPEN operations	Microseconds
FILEOPEN_OPS	Number of FILEOPEN operation per second	Ops/Sec
FLOCK_LATENCY	Average latency of FLOCK operations	Microseconds
FLOCK_OPS	Number of FLOCK operation per second	Ops/Sec
GETATTR_LATENCY	Average latency of GETATTR operations	Microseconds
GETATTR_OPS	Number of GETATTR operation per second	Ops/Sec
GETXATTR_LATENCY	Average latency of GETXATTR operations	Microseconds
GETXATTR_OPS	Number of GETXATTR operation per second	Ops/Sec
IOCTL_OBS_PREFETCH_LATENCY	Average latency of IOCTL_OBS_PREFETCH operations	Microseconds
IOCTL_OBS_PREFETCH_OPS	Number of IOCTL_OBS_PREFETCH operation per second	Ops/Sec
IOCTL_OBS_RELEASE_LATENCY	Average latency of IOCTL_OBS_RELEASE operations	Microseconds

Type	Description	Units
IOCTL_OBS_RELEASE_OPS	Number of IOCTL_OBS_RELEASE operation per second	Ops/Sec
LINK_LATENCY	Average latency of LINK operations	Microseconds
LINK_OPS	Number of LINK operation per second	Ops/Sec
LISTXATTR_LATENCY	Average latency of LISTXATTR operations	Microseconds
LISTXATTR_OPS	Number of LISTXATTR operation per second	Ops/Sec
LOOKUP_LATENCY	Average latency of LOOKUP operations	Microseconds
LOOKUP_OPS	Number of LOOKUP operation per second	Ops/Sec
MKNOD_LATENCY	Average latency of MKNOD operations	Microseconds
MKNOD_OPS	Number of MKNOD operation per second	Ops/Sec
OPS	Total number of operations	Ops/Sec
RDMA_WRITE_REQUESTS	Number of RDMA write request operations per second	Ops/Sec
READDIR_LATENCY	Average latency of READDIR operations	Microseconds
READDIR_OPS	Number of READDIR operation per second	Ops/Sec
READLINK_LATENCY	Average latency of READLINK operations	Microseconds
READLINK_OPS	Number of READLINK operation per second	Ops/Sec
READS	Number of read operations per second	Ops/Sec
READ_BYTES	Number of bytes read per second	Bytes/Sec

Type	Description	Units
READ_CHECKSUM_ERRORS		Ops
READ_DURATION		Microseconds
READ_LATENCY_NO_QOS	Average latency of READ operations without QoS delay	Microseconds
READ_LATENCY	Average latency of READ operations	Microseconds
READ_QOS_DELAY	Average QoS delay for READ operations	Microseconds
READ_RDMA_SIZES_RATE		Blocks/Sec
READ_RDMA_SIZES		Blocks
READ_SIZES_RATE		Blocks/Sec
READ_SIZES		Blocks
RENAME_LATENCY	Average latency of RENAME operations	Microseconds
RENAME_OPS	Number of RENAME operation per second	Ops/Sec
REQUESTS_COMPLETED		Ops
REQUESTS_FETCHED		Ops
RMDIR_LATENCY	Average latency of RMDIR operations	Microseconds
RMDIR_OPS	Number of RMDIR operation per second	Ops/Sec
RMXATTR_LATENCY	Average latency of RMXATTR operations	Microseconds
RMXATTR_OPS	Number of RMXATTR operation per second	Ops/Sec
SETATTR_LATENCY	Average latency of SETATTR operations	Microseconds
SETATTR_OPS	Number of SETATTR operation per second	Ops/Sec
SETXATTR_LATENCY	Average latency of SETATTR operations	Microseconds

Type	Description	Units
SETXATTR_OPS	Number of SETXATTR operation per second	Ops/Sec
STATFS_LATENCY	Average latency of STATFS operations	Microseconds
STATFS_OPS	Number of STATFS operation per second	Ops/Sec
SUCCEEDED_1HOP_READS	Number of succesfull single hop reads per second	Ops/Sec
SYMLINK_LATENCY	Average latency of SYMLINK operations	Microseconds
SYMLINK_OPS	Number of SYMLINK operation per second	Ops/Sec
THROUGHPUT	Number of byte read/writes per second	Bytes/Sec
UNLINK_LATENCY	Average latency of UNLINK operations	Microseconds
UNLINK_OPS	Number of UNLINK operation per second	Ops/Sec
WRITES	Number of write operations per second	Ops/Sec
WRITE_BYTE	Number of byte writes per second	Bytes/Sec
WRITE_DURATION		Microseconds
WRITE_LATENCY_NO_QOS	Average latency of WRITE operations without QoS delay	Microseconds
WRITE_LATENCY	Average latency of WRITE operations	Microseconds
WRITE_QOS_DELAY	Average QoS delay for WRITE operations	Microseconds
WRITE_RDMA_SIZES_RATE		Blocks/Sec
WRITE_RDMA_SIZES		Blocks/
WRITE_SIZES_RATE		Blocks/Sec

Type	Description	Units
WRITE_SIZES		Blocks/Sec

Operations

Type	Description	Units
ACCESS_LATENCY	Average latency of ACCESS operations	Microseconds
ACCESS_OPS	Number of ACCESS operation per second	Ops/Sec
COMMIT_LATENCY	Average latency of COMMIT operations	Microseconds
COMMIT_OPS	Number of COMMIT operation per second	Ops/Sec
CREATE_LATENCY	Average latency of CREATE operations	Microseconds
CREATE_OPS	Number of CREATE operation per second	Ops/Sec
FILEATOMICOPEN_LATENCY	Average latency of FILEATOMICOPEN operations	Microseconds
FILEATOMICOPEN_OPS	Number of FILEATOMICOPEN operation per second	Ops/Sec
FILECLOSE_LATENCY	Average latency of FILECLOSE operations	Microseconds
FILECLOSE_OPS	Number of FILECLOSE operation per second	Ops/Sec
FILEOPEN_LATENCY	Average latency of FILEOPEN operations	Microseconds
FILEOPEN_OPS	Number of FILEOPEN operation per second	Ops/Sec
FLOCK_LATENCY	Average latency of FLOCK operations	Microseconds

Type	Description	Units
FLOCK_OPS	Number of FLOCK operation per second	Ops/Sec
FSINFO_LATENCY	Average latency of FSINFO operations	Microseconds
FSINFO_OPS	Number of FSINFO operation per second	Ops/Sec
GETATTR_LATENCY	Average latency of GETATTR operations	Microseconds
GETATTR_OPS	Number of GETATTR operation per second	Ops/Sec
LINK_LATENCY	Average latency of LINK operations	Microseconds
LINK_OPS	Number of LINK operation per second	Ops/Sec
LOOKUP_LATENCY	Average latency of LOOKUP operations	Microseconds
LOOKUP_OPS	Number of LOOKUP operation per second	Ops/Sec
MKDIR_LATENCY	Average latency of MKDIR operations	Microseconds
MKDIR_OPS	Number of MKDIR operation per second	Ops/Sec
MKNOD_LATENCY	Average latency of MKNOD operations	Microseconds
MKNOD_OPS	Number of MKNOD operation per second	Ops/Sec
OPS	Total number of operations	Ops/Sec
PATHCONF_LATENCY	Average latency of PATHCONF operations	Microseconds
PATHCONF_OPS	Number of PATHCONF operation per second	Ops/Sec
READDIR_LATENCY	Average latency of READDIR operations	Microseconds
READDIR_OPS	Number of READDIR operation per second	Ops/Sec

Type	Description	Units
READLINK_LATENCY	Average latency of READLINK operations	Microseconds
READLINK_OPS	Number of READLINK operation per second	Ops/Sec
READS	Number of read operations per second	Ops/Sec
READ_BYTES	Number of bytes read per second	Bytes/Sec
READ_DURATION		Microseconds
READ_LATENCY	Average latency of READ operations	Microseconds
REMOVE_LATENCY	Average latency of READ operations	Microseconds
REMOVE_OPS	Number of REMOVE operation per second	Ops/Sec
RENAME_LATENCY	Average latency of RENAME operations	Microseconds
RENAME_OPS	Number of RENAME operation per second	Ops/Sec
RMDIR_LATENCY	Average latency of RMDIR operations	Microseconds
RMDIR_OPS	Number of RMDIR operation per second	Ops/Sec
SETATTR_LATENCY	Average latency of SETATTR operations	Microseconds
SETATTR_OPS	Number of SETATTR operation per second	Ops/Sec
STATFS_LATENCY	Average latency of STATFS operations	Microseconds
STATFS_OPS	Number of STATFS operation per second	Ops/Sec
SYMLINK_LATENCY	Average latency of SYMLINK operations	Microseconds
SYMLINK_OPS	Number of SYMLINK operation per second	Ops/Sec

Type	Description	Units
THROUGHPUT	Number of byte read/writes per second	Bytes/Sec
UNLINK_LATENCY	Average latency of UNLINK operations	Microseconds
UNLINK_OPS	Number of UNLINK operation per second	Ops/Sec
WRITES	Number of write operations per second	Ops/Sec
WRITE_BYTES	Number of byte writes per second	Bytes/Sec
WRITE_DURATION		Microseconds
WRITE_LATENCY	Average latency of WRITE operations	Microseconds

RAFT

Type	Description	Units
Bucket_LEADER_CHANGES	Changes of leader	Changes
Bucket_REQUESTS_COMPLETED	Requests to leader completed successfully	Requests
Configuration_LEADER_CHANGES	Changes of leader	Changes
Configuration_REQUESTS_COMPLETED	Requests to leader completed successfully	Requests
Invalid_LEADER_CHANGES	Changes of leader	Changes
Invalid_REQUESTS_COMPLETED	Requests to leader completed successfully	Requests
SYNCLOG_TIMEOUTS	Number of times timeouted on syncing logs to node	Timeouts
Test_LEADER_CHANGES	Changes of leader	Changes
Test_REQUESTS_COMPLETED	Requests to leader completed successfully	Requests

RAID

Type	Description	Units
LONG_RPC_TIMEOUTS	Long RPC timeouts encountered	Occurences
RAID_BLOCKS_IN_PREPARED_STRIPE	Free blocks in prepared stripe	
RAID_CHUNKS_CLEANED_BY_SHIFT	Dirty chunks cleaned by being shifted out	Occurences
RAID_CHUNKS_SHIFTED	Dirty chunks that shifted out	Occurences
RAID_COMMITTED_STRIPES	Num stripes written	Stripes
RAID_PLACEMENT_SWITCHES	Num placement switches	Switches
RAID_READ_BATCHES_PER_REQUEST_HISTOGRAM	Histogram of number of batches of stripes read in a single request	
RAID_READ_BLOCKS_STRIPE_HISTOGRAM	Histogram of number of blocks read from a single stripe	
RAID_READ_BLOCKS	Number of blocks read by the RAID	Blocks/Sec
RAID_READ_DEGRADED	Degraded mode reads	Blocks/Sec
RAID_READ_IOS	Raw read blocks performed by the RAID	Blocks/Sec
RAID_STALE_WRITES_DETECTED	Stale write detected in read	Occurences

RPC

Type	Description	Units
CLIENT_CANCELED_REQUESTS		Calls/Sec
CLIENT_DROPPED_RESPONSES		Calls/Sec

Type	Description	Units
CLIENT_RECEIVED_EXCEPTIONS		Calls/Sec
CLIENT_RECEIVED_RESPONSES		Calls/Sec
CLIENT_RECEIVED_TIMEOUTS		Calls/Sec
CLIENT_ROUNDTRIP_AVG_LOW		Microseconds
CLIENT_ROUNDTRIP_AVG		Microseconds
CLIENT_RPC_CALLS_LOW		RPC/Sec
CLIENT_RPC_CALLS		RPC/Sec
CLIENT_SENT_REQUESTS		Calls/Sec
FIRST_RESULTS	Number of first results per second	Ops/Sec
SERVER_ABORTS		Calls/Sec
SERVER_DROPPED_REQUESTS		Calls/Sec
SERVER_PROCESSING_AVG		Microseconds
SERVER_PROCESSING_TIME		
SERVER_RECEIVED_REQUESTS		Calls/Sec
SERVER_REJECTS		Calls/Sec
SERVER_RPC_CALLS		RPC/Sec
SERVER_SENT_EXCEPTIONS		Calls/Sec
SERVER_SENT_RESPONSES		Calls/Sec
TIME_TO_FIRST_RESULT	Average latency to the first result of a MultiCall	Microseconds

Reactor

Type	Description	Units
BACKGROUND_CYCLES	Number of cycles spent in background fibers	Cycles/Sec
BACKGROUND_FIBERS	Number of background fibers that are ready to run and eager to get CPU cycles	Fibers
BACKGROUND_TIME	Percentage of the CPU time utilized for background operations	%
BucketInvocationState_CAPACITY	Number of data structures allocated to the BucketInvocationState pool	Structs
BucketInvocationState_STRUCT_SIZE	Number of bytes in each struct of the BucketInvocationState pool	Bytes
BucketInvocationState_USED	Number of structs in the BucketInvocationState pool which are currently being used	Structs
Bucket_CAPACITY	Number of data structures allocated to the Bucket pool	Structs
Bucket_STRUCT_SIZE	Number of bytes in each struct of the Bucket pool	Bytes
Bucket_USED	Number of structs in the Bucket pool which are currently being used	Structs
CLASS_BLOB!(RAID)_CAPACITY	Number of data structures allocated to the CLASS_BLOB!(RAID) pool	Structs
CLASS_BLOB!(RAID)_STRUCT_SIZE	Number of bytes in each struct of the CLASS_BLOB!(RAID) pool	Bytes
CLASS_BLOB!(RAID)_USED	Number of structs in the CLASS_BLOB!(RAID) pool which are currently being used	Structs

Type	Description	Units
CYCLES_PER_SECOND	Number of cycles the cpu runs per second	Cycles/Sec
ChainedSpan_CAPACITY	Number of data structures allocated to the ChainedSpan pool	Structs
ChainedSpan_STRUCT_SIZE	Number of bytes in each struct of the ChainedSpan pool	Bytes
ChainedSpan_USED	Number of structs in the ChainedSpan pool which are currently being used	Structs
Charter_CAPACITY	Number of data structures allocated to the Charter pool	Structs
Charter_STRUCT_SIZE	Number of bytes in each struct of the Charter pool	Bytes
Charter_USED	Number of structs in the Charter pool which are currently being used	Structs
CrossDestageDesc_CAPACITY	Number of data structures allocated to the CrossDestageDesc pool	Structs
CrossDestageDesc_STRUCT_SIZE	Number of bytes in each struct of the CrossDestageDesc pool	Bytes
CrossDestageDesc_USED	Number of structs in the CrossDestageDesc pool which are currently being used	Structs
DEFUNCT_FIBERS	Number of defunct buffers, which are really just memory structures allocated for future fiber needs.	Fibers
DeferredTask2_CAPACITY	Number of data structures allocated to the DeferredTask2 pool	Structs
DeferredTask2_STRUCT_SIZE	Number of bytes in each struct of the DeferredTask2 pool	Bytes

Type	Description	Units
DeferredTask2_USED	Number of structs in the DeferredTask2 pool which are currently being used	Structs
EXCEPTIONS	Number of execptions caught by the reactor	Exceptions/Sec
GenericBaseBlock_CAPACITY	Number of data structures allocated to the GenericBaseBlock pool	Structs
GenericBaseBlock_STRUCT_SIZE	Number of bytes in each struct of the GenericBaseBlock pool	Bytes
GenericBaseBlock_USED	Number of structs in the GenericBaseBlock pool which are currently being used	Structs
HOGGED_TIME	Histogram of time used by hogger fibers (only in debug builds)	
IDLE_CALLBACK_INVOCATIONS	Number of background work invocations	Invocations/Sec
IDLE_CYCLES	Number of cycles spent in idle	Cycles/Sec
IDLE_TIME	Percentage of the CPU time not utilized for handling I/Os	%
NODE_HANG		
OUTRAGEOUS_HOGGERS	Number of hoggers taking really excessive amount of time to run	Invocations
ObsGateway_CAPACITY	Number of data structures allocated to the ObsGateway pool	Structs
ObsGateway_STRUCT_SIZE	Number of bytes in each struct of the ObsGateway pool	Bytes
ObsGateway_USED	Number of structs in the ObsGateway pool which are currently being used	Structs

Type	Description	Units
PENDING_FIBERS	Number of fibers pending for external events, such as a network packet, or SSD response. Upon such external event they will change state to scheduled fibers	Fibers
QueuedBlock_CAPACITY	Number of data structures allocated to the QueuedBlock pool	Structs
QueuedBlock_STRUCT_SIZE		Bytes
QueuedBlock_USED	Number of bytes in each struct of the QueuedBlock pool	Structs
ReadBlocksImpl!(RAID)_CAPACITY	Number of data structures allocated to the ReadBlocksImpl!(RAID) pool	Structs
ReadBlocksImpl!(RAID)_STRUCT_SIZE	Number of bytes in each struct of the ReadBlocksImpl!(RAID) pool	Bytes
ReadBlocksImpl!(RAID)_USED	Number of structs in the ReadBlocksImpl!(RAID) pool which are currently being used	Structs
SCHEDULED_FIBERS	Number of current fibers that are ready to run and eager to get CPU cycles	Fibers
SLEEPY_FIBERS	Number of SLEEPY fibers	Sleepy fiber detections
SLEEPY_RPC_SERVER_FIBERS	Number of SLEEPY RPC server fibers	Structs
SSD_CAPACITY	Number of data structures allocated to the SSD pool	Structs
SSD_STRUCT_SIZE	Number of bytes in each struct of the SSD pool	Bytes
SSD_USED	Number of structs in the SSD pool which are currently being used	Structs

Type	Description	Units
STEP_CYCLES	Histogram of time spent in a fiber	
TIMER_CALLBACKS	Current number of timer callbacks	Callbacks
TOTAL_FIBERS_COUNT	Number of fibers	Fibers
TimedCallback_CAPACITY	Number of data structures allocated to the TimedCallback pool	Structs
TimedCallback_STRUCT_SIZE	Number of bytes in each struct of the TimedCallback pool	Bytes
TimedCallback_USED	Number of structs in the TimedCallback pool which are currently being used	Structs
UploadFileInfo_CAPACITY	Number of data structures allocated to the UploadFileInfo pool	Structs
UploadFileInfo_STRUCT_SIZE	Number of bytes in each struct of the UploadFileInfo pool	Bytes
UploadFileInfo_USED	Number of structs in the UploadFileInfo pool which are currently being used	Structs
networkBuffers_CAPACITY	Number of data structures allocated to the networkBuffers pool	Structs
networkBuffers_USED	Number of structs in the networkBuffers pool which are currently being used	Structs
rdmaNetworkBuffers_CAPACITY	Number of data structures allocated to the rdmaNetworkBuffers pool	Structs
rdmaNetworkBuffers_USED	Number of structs in the rdmaNetworkBuffers pool which are currently being used	Structs

SSD

Type	Description	Units
DRIVE_ACTIVE_IOS	The number of in flight IO against the SSD at the time of sampling	IOs
DRIVE_FORFEITS	Number of IOs forfeited due to lack of memory buffers	Operations/Sec
DRIVE_IDLE_CYCLES	Number of cycles spent in idle	Cycles/Sec
DRIVE_IDLE_TIME	Percentage of the CPU time not utilized for handling I/Os	%
DRIVE_IO_OVERLAPPED	Number of overlapping IOs	Operations
DRIVE_IO_TOO_LONG	Number of IOs that took longer than expected	Operations/Sec
DRIVE_LATENCY	Measure the latencies up to 5ms (higher latencies will be grouped together)	
DRIVE_LOAD	Drive Load at sampling time	Load
DRIVE_MEDIA_BLOCKS_READ	Blocks read from the SSD media	Blocks/Sec
DRIVE_MEDIA_BLOCKS_WRITE	Blocks written to the SSD media	Blocks/Sec
DRIVE_MEDIA_ERRORS	SSD Media Errors	IO/Sec
DRIVE_NON_MEDIA_ERRORS	SSD Non-Media Errors	IO/Sec
DRIVE_PENDING_IOS	The number of IOs waiting to start executing at the time of sampling	IO
DRIVE_PUMPED_IOS	Number of requests returned in a pump	Operations/Sec
DRIVE_PUMPS_DELAYED	Number of Drive pumps that got delayed	Operations/Sec
DRIVE_PUMPS_SEVERELY_DELAYED	Number of Drive pumps that got severely delayed	Microseconds
DRIVE_PUMP_LATENCY	Latency between SSD pumps	Microseconds

Type	Description	Units
DRIVE_READ_LATENCY	Drive Read Execution Latency	Microseconds
DRIVE_READ_OPS	Drive Read Operations	IO/Sec
DRIVE_REMAINING_IOS	Number of requests still in the drive after a pump	
DRIVE_REQUEST_BLOCKS	Measure drive request size distribution	
DRIVE_SSD_PUMPS	Number of drive pumps that resulted in data flowin from/to drive	Pump/Sec
DRIVE_UTILIZATION	Percentage of time the drive had an active IO submitted to it	%
DRIVE_WRITE_LATENCY	Drive Write Execution Latency	Microseconds
DRIVE_WRITE_OPS	Drive Write Operations	IO/Sec
SSDS_IOS	IOs performed on the SSD service	IO/Sec
SSDS_IO_ERRORS	IO errors on the SSD service	Blocks/Sec
SSD_BLOCKS_READ	Number of blocks read from the SSD service	Blocks/Sec
SSD_BLOCKS_WRITTEN	Number of blocks written to the SSD service	Blocks/Sec
SSD_CHUNK_ALLOCS	Rate of chunk allocations	Chunks/Sec
SSD_CHUNK_FREES	Rate of chunk frees	Chunks/Sec
SSD_E2E_BAD_CSUM	Errors in reading blocks from the SSD service	IO/Sec
SSD_READ_ERRORS	Errors in reading blocks from the SSD service	Blocks/Sec
SSD_READ_LATENCY	Avg. latency of read requests from the SSD service	Microseconds
SSD_READ_REQS_LARGE_NORMAL	Number of large normal read requests from the SSD service	IO/Sec

Type	Description	Units
SSD_READ_REQS	Number of read requests from the SSD service	IO/Sec
SSD_WRITES_REQS_LARGE_NORMAL	Number of large normal priority write requests to the SSD service	IO/Sec
SSD_WRITES	Number of write requests to the SSD service	IO/Sec
SSD_WRITE_ERRORS	Errors in writing blocks to the SSD service	Blocks/Sec
SSD_WRITE_LATENCY	Latency of writes to the SSD service	Microseconds

Scrubber

Type	Description	Units
ACTUALLY_FALSE_FREE	Number of blocks that were detected as false-used and freed	Blocks/Sec
CLEANED_CHUNKS	Number of chunks that were cleaned by the scrubber	Chunks/Sec
DEGRADED_READS	Number of degraded reads for scrubbing	Requests/Sec
FALSE_FREE_CHECKED_BLOCKS	Number of blocks that were scrubbed-false-used	Blocks/Sec
FALSE_FREE_CHECK_LATENCY	Average latency of checking false free per block	Micros
FALSE_USED_CHECKED_BLOCKS	Number of blocks that were scrubbed-false-used	Blocks/Sec
FALSE_USED_CHECK_LATENCY	Average latency of checking false used per block	Micros
FALSE_USED_EXTRA_NOTIFIED	Number of blocks that were notified as used by the mark-extra-used mechanism	Blocks/Sec
INTERRUPTS	Number of scrubs that were interrupted	Occurences/Sec

Type	Description	Units
NETWORK_BUDGET_WAIT_LATENCY	Average latency of waiting for our network budget	Micros
NOT_ACTUALLY_FALSE_FREE	Number of blocks that were detected as used	Blocks/Sec
NOT_REALLY_DIRTY_BLOCKS	Number of marked dirty blocks that ScrubMissingWrites found were actually clean	Blocks/Sec
NUM_COPY_DISCARDED_BLOCKS	Number of copied blocks that were discarded	Blocks/Sec
NUM_COPY_DISCARDS	Number of times we discarded scrubber copy work	Occurences/Sec
NUM_INVENTED_STRIPES_DISCARDS	Number of times we discarded all scrubber work due to invented stripes	Occurences/Sec
NUM_INVENTED_STRIPES_DISCARD_BLOCKS	Number of blocks that were discarded due to invented stripes	Blocks/Sec
NUM_SCRUBBER_DISCARD_INTERMEDIATES	Number of times we discarded all intermediate scrubber work	Occurences/Sec
NUM_SMW_DISCARDED_BLOCKS	Number of SMW'd blocks that were discarded	Blocks/Sec
NUM_SMW_DISCARDS	Number of times we discarded scrubber SMW work	Occurences/Sec
PLACEMENT_SELECTION_LATENCY	Average latency of scrubbed placement selection	Micros
READS_CALLED	Number of blocks that were read	Blocks/Sec
READ_BATCH_SOURCE_BLOCKS	Number of source blocks to read in batch	
READ_BLOCKS_LATENCY	Average latency of read blocks	Micros
RELOCATED_BLOCKS	Number of blocks that were relocated for eviction	Blocks/Sec

Type	Description	Units
RELOCATE_BLOCKS_LATENCY	Average latency of relocating blocks	Micros
RETRUSTED_UNPROTECTED_DIRTY_BLOCKS	Number of dirty blocks that ScrubMissingWrites retrusted because they were unprotected	Blocks/Sec
REWRITTEN_DIRTY_BLOCKS	Number of dirty blocks that ScrubMissingWrites rewrote to clean them	Blocks/Sec
SCRUB_BATCHES_LATENCY	Average latency of scrub batches	Millis
SCRUB_FALSE_FREE_FAILED_READS	Number of blocks that we failed to read for scrub-false-free	Blocks/Sec
SCRUB_FALSE_FREE_FAILED	Number of placements we failed to fully scrub-false-free	Occurences/Sec
SCRUB_FALSE_FREE_PLACEMENTS	Number of placements we finished scrub-false-used	Occurences/Sec
SCRUB_FALSE_FREE_WAS_UNPROTECTED	Number of blocks that were false marked freed and unprotected	Blocks/Sec
SCRUB_FALSE_USED_FAILED_READS	Number of blocks that we failed to read for scrub-false-used	Blocks/Sec
SCRUB_FALSE_USED_FAILED	Number of placements we failed to fully scrub-false-used	Occurences/Sec
SCRUB_FALSE_USED_PLACEMENTS	Number of placements we finished scrub-false-used	Occurences/Sec
SCRUB_FALSE_USED_WAS_UNPROTECTED	Number of blocks that were false marked used and unprotected	Blocks/Sec
SCRUB_PREPARATION_FAILED	Number of times we failed to prepare() a task and aborted scrub of placement	Occurences/Sec
SFU_CHECKS	Number of blocks that were scrubbed-false-used	Blocks/Sec

Type	Description	Units
SFU_CHECK_FREE	Number of blocks that were detected as false-used and freed	Blocks/Sec
SFU_CHECK_SECONDARY	Number of blocks that were detected as secondary	Blocks/Sec
SFU_CHECK_USED	Number of blocks that were detected as used	Blocks/Sec
SOURCE_READS	Number of source/committed superset blocks directly read by the scrubber	Blocks/Sec
TARGET_COPIED_CHUNKS	Number of chunks that were copied to target by the scrubber	Chunks/Sec
UPDATE_PLACEMENT_INFO_LATENCY	Average latency of updating the placement info quorum	Micros
UPDATE_PLACEMENT_INFO	Number of times we ran updatePlacementInfo	Occurences/Sec
WONT_CLEAN_COPYING	Number of actually dirty blocks that ScrubMissingWrites refused to clean because they will be moved to target anyway	Blocks/Sec
WRITES_CALLED	Number of blocks that were written	Blocks/Sec
WRITE_BATCH_SOURCE_BLOCKS	Number of source blocks to write in batch	
WRITE_BATCH_TARGET_BLOCKS	Number of target blocks to write in batch	
WRITE_BLOCKS_LATENCY	Average latency of writing blocks	Micros

Squelch

Type	Description	Units
BLOCKS_PER_DESQUELCH		Blocks

Type	Description	Units
EXTENT_DESQUELCHES_NUM		Times
EXTENT_SQUELCH_BLOCKS_READ		Blocks
HASH_DESQUELCHES_NUM		Times
HASH_SQUELCH_BLOCKS_READ		Blocks
INODE_DESQUELCHES_NUM		Times
INODE_SQUELCH_BLOCKS_READ		Blocks
JOURNAL_DESQUELCHES_NUM		Times
JOURNAL_SQUELCH_BLOCKS_READ		Blocks
MAX_BLOCKS_WITH_TEMPORAL_SQUELCH_ITEMS_IN_BUCKET	Number of block with temporal squelch items in bucket	Blocks
MAX_TEMPORAL_SQUELCH_ITEMS_IN_BUCKET		Squelch items
REGISTRY_L1_DESQUELCHES_NUM		Times
REGISTRY_L1_SQUELCH_BLOCKS_READ		Blocks
REGISTRY_L2_DESQUELCHES_NUM		Times
REGISTRY_L2_SQUELCH_BLOCKS_READ		Blocks
SPATIAL_SQUELCH_DESQUELCHES_NUM		Times
SPATIAL_SQUELCH_SQUELCH_BLOCKS_READ		Blocks
SUPERBLOCK_DESQUELCHES_NUM		Times

Type	Description	Units
SUPERBLOCK_SQUELCH_BLOCKS_READ		Blocks
TEMPORAL_SQUELCH_DESQUELCHES_NUM		Times
TEMPORAL_SQUELCH_SQUELCH_BLOCKS_READ		Blocks

Statistics

Type	Description	Units
GATHER_FROM_NODE_LATENCY_NET	Time spent on responding to a stats gathering request(not including metadata)	Seconds/Sec
GATHER_FROM_NODE_LATENCY	Time spent responding to a stats gathering request(not including metadata)	Seconds/Sec
GATHER_FROM_NODE_SLEEP	Time spent in-between responding to a stats gathering request(not including metadata)	Seconds/Sec
TIMES_QUERIED_STATS		Times
TIMES_QUERIED	Number of times the node was queried for stats (not including metadata)	Times

Chapter 16: System congestion

Possible congestion issues in the Content Software for File system are described.

Overview

The Content Software for File system is built to be very efficient, provide maximum performance and saturate the network links.

In some situations, the system itself may slow down IOs when reaching some limits (or even block new IOs at higher limits) until the congested resource is relieved. Such situations may be transient and the issue will be resolved on its own after a short time. However, there are also cases that suggest an issue that needs to be addressed, such as a workload maxing out the resources of the cluster. In such cases, the cluster resources must be expanded, as described in Expanding and shrinking cluster resources (on page 239). For more information contact customer support.

System congestion events or alerts

The Content Software for File system can issue several types of congestion events/alerts:

Type	Description	Actions
FIBERS	Extreme load of concurrent system operations on a node.	This is usually a transient situation due to load on them system. If the load is consistent and the problem persists, add more resources (hosts/cores), as described in Adding a backend host or Addition of CPUcores.
DESTAGER	Too many pending IOs are waiting to be written for a specific node.	This is usually a transient situation due to load on them system. If the load is consistent and the problem persists, add more hosts to the cluster as described in Adding a backend host or expand the host resources as described in "Expansion of specific resources" in the *Content Software for File Command Line Reference*.

Type	Description	Actions
SSD	Too many pending IOs to the SSD.	If there is a single SSD, it is probably faulty and needs to be replaced. If there are multiple SSDs, the load on the system is too high. To handle such a load, more SSDs should be added to the system, as described in Expansion of specific resources.
RAID_NOT_OK	More IO failures than can be handled have occurred, and IOs cannot be served.	Make sure to bring up any host that might be down. If all hosts are up, contact customer support.
XDESTAGE	Auxiliary cluster resources are low.	This is usually a transient situation due to load on the system. If the load is consistent and the problem persists, add more hosts to the cluster as described in Adding a backend host or contact customer support.

Chapter 17: Security management

This page describes important security consideration for the Content Software for File cluster management.

The Content Software for File system is a secured environment. It deploys a combination of security controls to ensure secured communication and secured user data.

The security controls include the following:

- HTTPS access: To access the Weka GUI, you connect only to one of the system servers using HTTPS through port 14000.
- Authentication tokens: The authentication tokens are used for accessing the Weka system API and to allow the mounting of secure filesystems.
- KMS: When creating an encrypted filesystem, a KMS must be used to properly secure the encryption keys. The KMS encrypts and decrypts filesystem keys.
- TLS certificates: By default, the system deploys a self-signed certificate to access the GUI, CLI, and API through HTTPS. You can deploy your certificate by providing an unencrypted private key and certificate PEM files.
- CA certificates: The system uses well-known CA certificates to establish trust with external services. For example, when using a KMS.
- Account lockout: To prevent brute force attacks, if several login attempts fail (default: 5), the user account is locked for several minutes (default: 2 minutes).
- Login banner: The login banner provides a security statement or a legal message displayed on the sign-in page.
- GUI session automatic termination: The user is signed out after 30 minutes of inactivity.

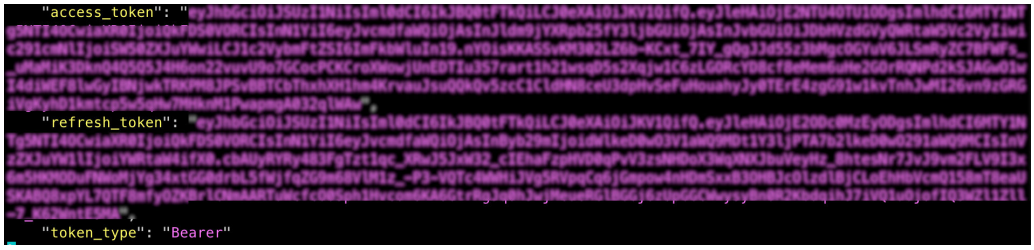
Obtaining authentication tokens

The authentication tokens include two types: an access token and a refresh token.

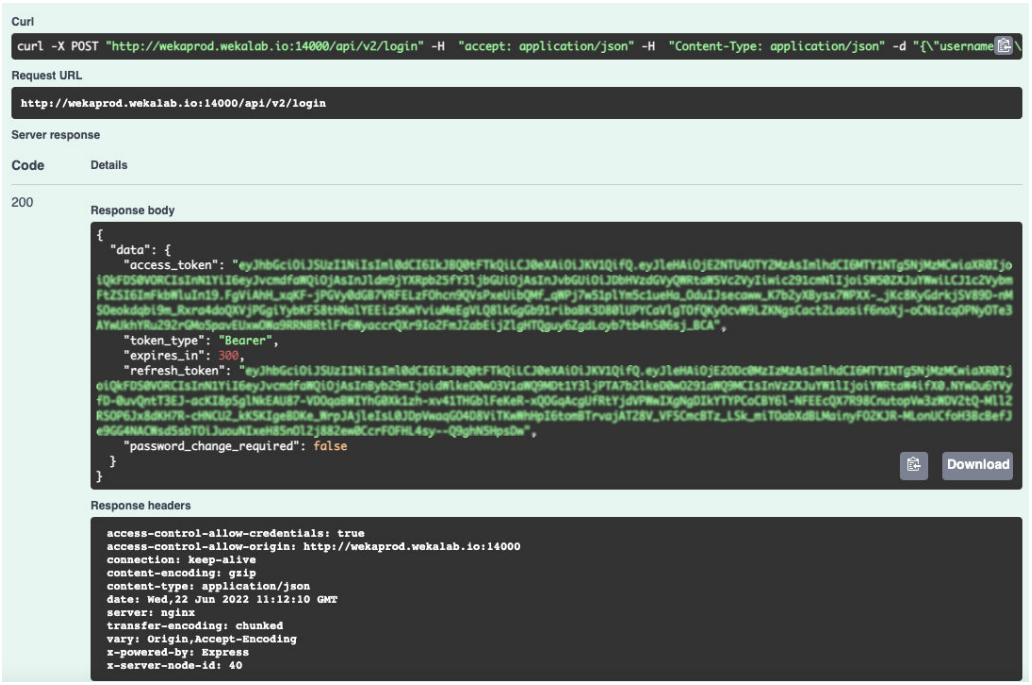
- Access token: The access token is a short-live token (five minutes) used for accessing the Weka system API and to allow the mounting of secure filesystems.
- Refresh token: The refresh token is a long-live token (one year) used for obtaining an additional access token.

Procedure

1. Do one of the following:
   - To obtain the refresh token and access token, through the CLI, log in to the system using the command: `weka user login`. The system creates an authentication token file and saves it in: `~/.weka/auth-token.json`. The token file contains both the access token and refresh token.



   - To obtain the refresh token and access token, through the REST API, use the `POST /login`. The API returns the token in the response body.



Generating an access token for API usage (for internal users only)

When working with the REST API, internal Weka users may require using a longer-lived access token (a token that doesn't require a refresh every 5 minutes).

You can generate a longer-lived access token using the CLI command:

```
weka user generate-token [--access-token-timeout timeout]
```

The default timeout is 30 days.

To revoke the access and refresh tokens, use the CLI command:

```
weka user revoke-tokens
```

KMS management

The management of a Key Management System (KMS) within the Content Software for File system is described.

Overview

When creating an encrypted filesystem, a KMS must be used to properly secure the encryption keys.

The Weka system uses the KMS to encrypt filesystem keys. When the Content Software for File system comes up, it uses the KMS to decrypt the filesystem keys and use its in-memory capabilities for data encrypting/decrypting operations.

When a snapshot is taken using the Snap-To-Object feature, the encrypted filesystem key is saved along with the encrypted data. In the event of rehydrating this snapshot to a different filesystem (or when recovering from a disaster to the same filesystem in the Content Software for File cluster), the KMS is used to decrypt the filesystem key. Consequently, the same KMS data must be present when performing such operations.

For increased security, the Content Software for File system does not save any information that can reconstruct the KMS encryption keys, which is performed by the KMS configuration alone. Therefore, the following should be considered:

1. If the KMS configuration is lost, the encrypted data may also be lost. Therefore, a proper DR strategy should be set when deploying the KMS in a production environment.
2. The KMS must be available when the Content Software for File system comes up when a new filesystem is created, and from time to time when key rotations must be performed. Therefore, it is recommended that the KMS be highly available.

For more information, refer to KMS Best Practices (on page 209).

The Content Software for File system supports the following KMS type:

- Key Management Interoperability Protocol (KMIP)-compliant KMS (protocol version 1.2 and up).

For additional information on KMS support, contact your Hitachi representative.

KMS best practices

The KMS is the only source holding the key to decrypt Content Software for File system filesystem keys. For non-disruptive operation, it is highly recommended to follow these guidelines:

- Set up DR for the KMS (backup/replication) to avoid any chance of data loss.
- Ensure that the KMS is highly available (note that the KMS is represented by a single URL in the Content Software for File system).

- Provide access to the KMS from the Content Software for File system backend hosts.
- Verify the methods used by the KMS being implemented (each KMS has different methods for securing/unsealing keys and for reconstructing lost keys, for example, Vault unsealing methods, which enable the configuration of auto unsealing using a trusted service).
- Refer to Production Hardening for additional best practices suggested by HashiCorp when using Vault.

Note: Taking a Snap-To-Object ensures that the (encrypted) filesystems keys are backed up to the object store, which is important if a total corruption of the Content Software for File system configuration occurs.

Managing KMS using the GUI

How to manage KMS using the GUI.

Adding a KMS

Procedure

1. From the menu, select **Configure > Cluster Settings**.
2. From the left pane, select **Security**.
3. On the **Security** page, select **Configure KMS**.
4. On the **Configure KMS** dialog, select the KMS type to deploy: **HashiCorp Vault** or **Kmip**.

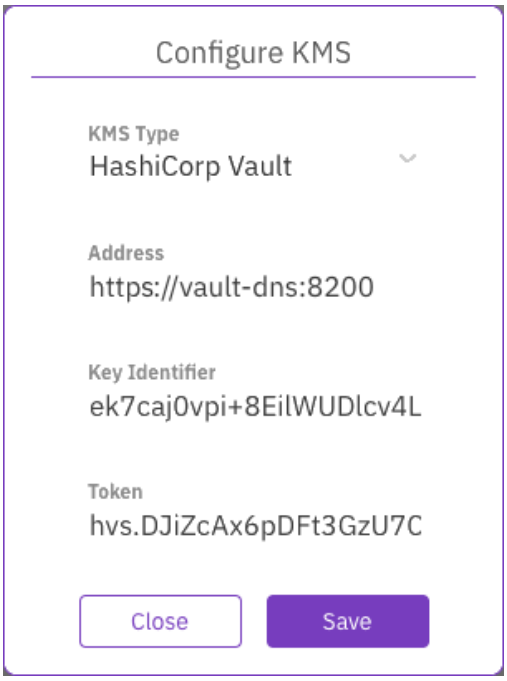


Figure 1 Configure KMS of HashiCorp Vault type

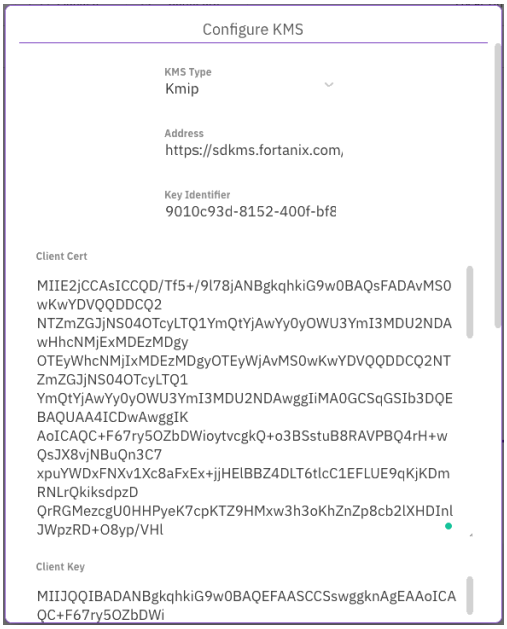


Figure 2 Configure KMS of Kmpi type

5. Enter the connection properties. The required properties depend on the KMS type you select.

   For the **HashiCorp Vault** type, enter the following:

   - **Address**: The KMS address
   - **Key Identifier**: The identifier of the KMS.
   - **Token**: The API token that you obtain from the vault.

For the **Kmip** type, enter the following:

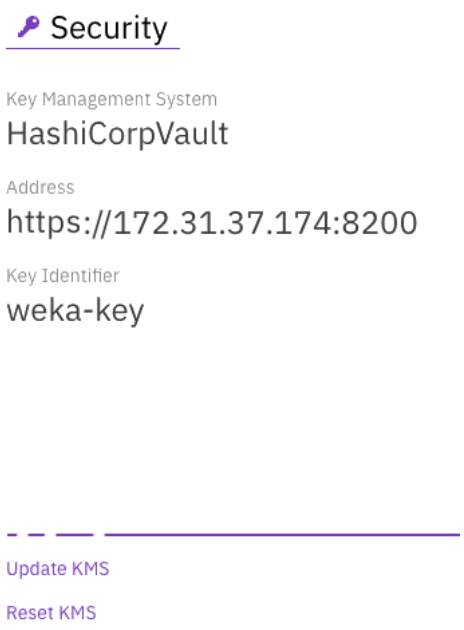
- **Address**: The KMS address
- **KMS Identifier**: The identifier of the KMS.
- **Client Cert** and **Client Key**: The client certificate and key that you obtain for the Kmip-based KMS.
- **CA Cert**: (Optional) A digital certificate from the Certificate Authority (CA).

6. Click **Save**.

Viewing the KMS

Procedure

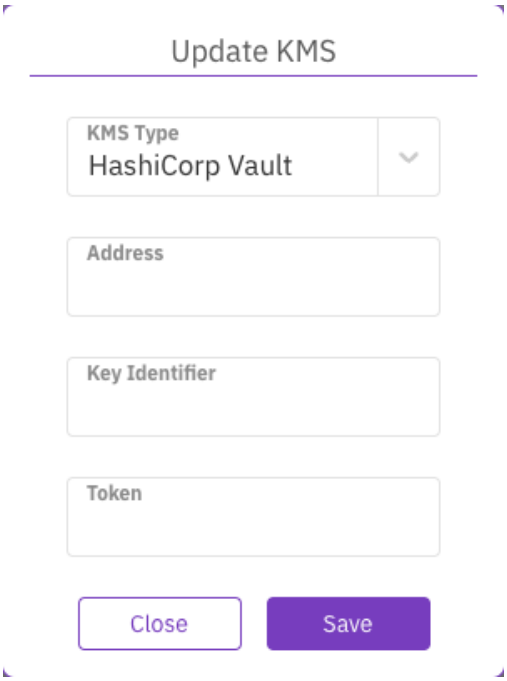
1. From the menu, select **Configure > Cluster Settings**.
2. From the left pane, select **Security**.
3. The **Security** page displays the configured KMS.



Updating the KMS configuration

Procedure

1. From the menu, select **Configure > Cluster Settings**.
2. From the left pane, select **Security**.
3. The **Security** page displays the configured KMS.
4. Select **Update KMS**, and update its settings.



5. Select **Save**.

Removing the KMS

Before you begin

Removing a KMS configuration is possible only if no encrypted filesystems exist.

Procedure

1. From the menu, select **Configure > Cluster Settings**.
2. From the left pane, select **Security**.
3. The **Security** page displays the configured KMS.
4. Select **Reset KMS**.
5. In the message that appears, select **Yes** to confirm the KMS configuration reset.

TLS certificate management

This page describes how manage the TLS certificate.

TLS certificates are used to protect both the clients' information while it's in transfer and to authenticate the system identity to ensure users are interacting with legitimate system owners.

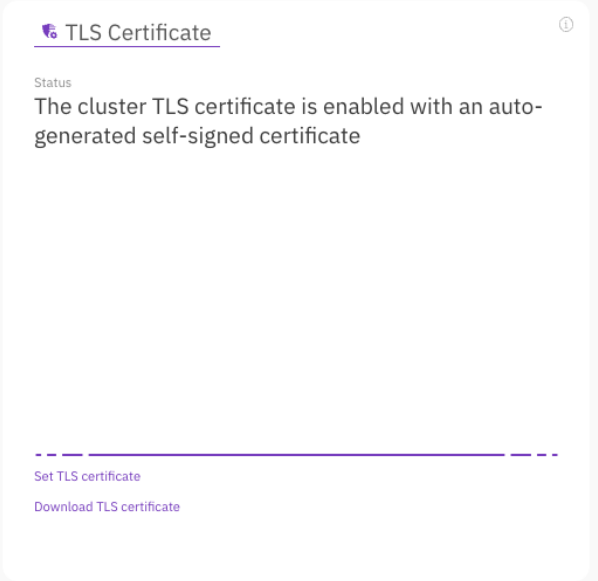
By default, the system deploys a self-signed certificate to access the GUI, CLI, and API through HTTPS. You can deploy your certificate by providing an unencrypted private key and certificate PEM files.

The system supports TLS 1.2 and higher with at least 128-bit ciphers.

Managing the TLS certificate using the GUI

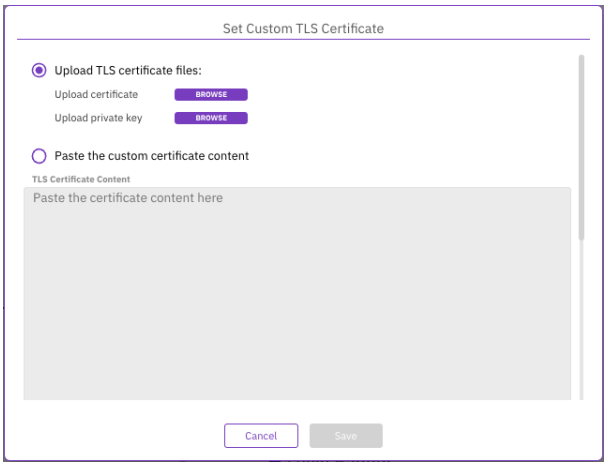
Once the system installation is completed, the cluster TLS certificate is enabled with an auto-generated self-signed certificate to access the GUI, CLI, and API through HTTPS. If you have a custom TLS certificate, you can set it instead of the auto-generated self-signed certificate.

You can also download the existing TLS certificate for later use if you want to use the self-signed certificate.

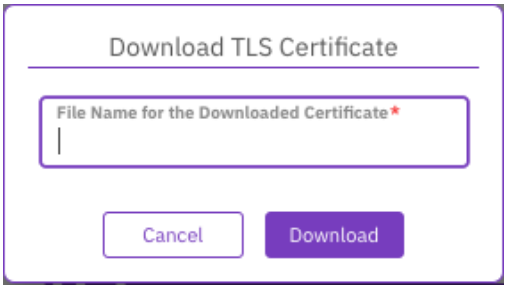


Procedure

1. From the menu, select **Configure > Cluster Settings**.
2. From the left pane, select **Security**.
3. In the TLS Certificate section, select **Set TLS certificate**.
4. In the Set Custom TLS Certificate dialog, do one of the following:
   - Select **Upload TLS certificate files**, and upload the TLS certificate and private key files.
   - Select **Paste the custom certificate content**, and paste the content of the TLS certificate and private key.



5. To download the existing TLS certificate, select **Download TLS certificate**. In the dialog, set a name for the certificate and select **Download**.



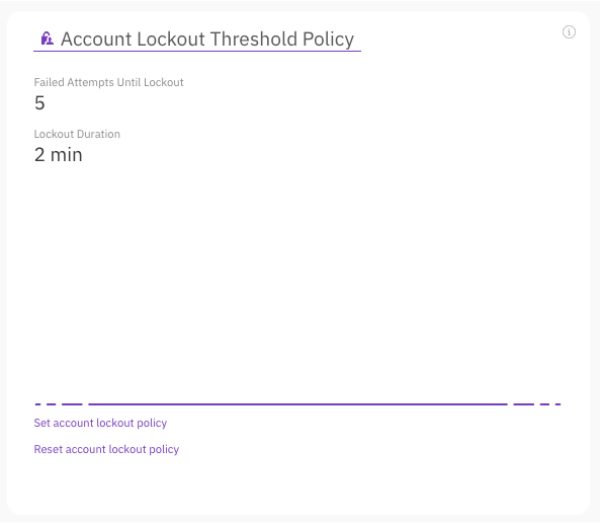
Account lockout threshold policy management

To prevent brute force attacks, if several sign-in attempts fail (default: 5), the user account is locked for several minutes (default: 2 minutes).

You can control these default values using the GUI or the CLI.

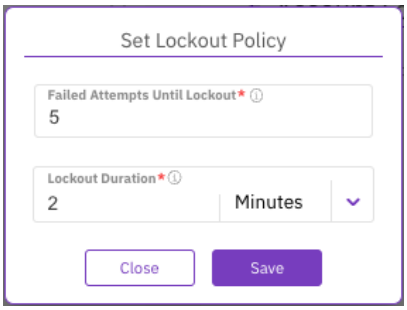
Manage the account lockout threshold policy using GUI

Using the GUI, you can set the number of failed attempts until the account is locked and the lockout duration. You can also reset the account lockout threshold policy properties to the default values



Procedure

1. From the menu, select **Configure > Cluster Settings**.
2. From the left pane, select **Security**.
3. In the Account Lockout Threshold Policy section, select Set Account Lockout Policy.
4. In the Set Lockout Policy dialog, do the following:
   - **Failed Attempts Until Lockout**: Set the number of sign-in attempts to lockout between 2 (minimum) to 50 (maximum).
   - **Lockout Duration**: Set the lockout duration between 30 secs (minimum) to 60 minutes (maximum).



5. Select Save.
6. To reset the account lockout threshold policy properties to the default values, select **Reset account lockout policy**. In the confirmation message, select **Yes**.

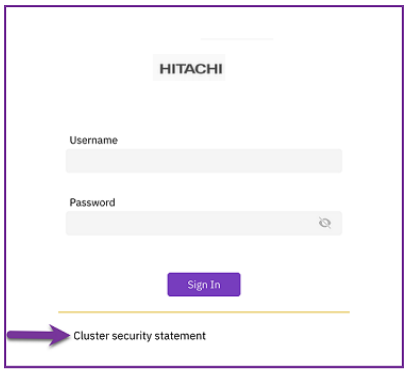
Managing the login banner

How to set a login banner displayed on the sign-in page.

The login banner provides a security statement or a legal message displayed on the sign-in page displayed on the GUI. The statement can be a definitive warning to any possible intruders that may want to access your system that certain types of activity are illegal, but at the same time, it also advises the authorized and legitimate users of their obligations relating to acceptable use of the system.

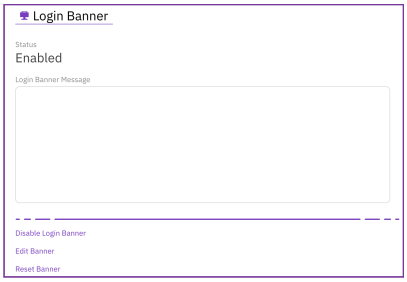
Managing the login banner using the GUI

You can set a login banner containing a security statement or a legal message displayed on the sign-in page. You can also disable, edit, or reset the login banner.

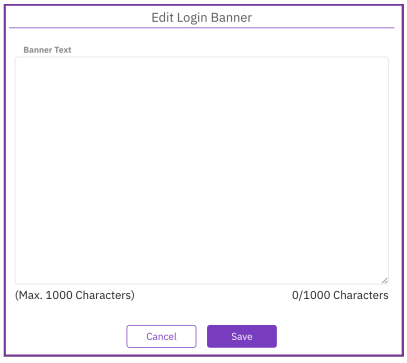


Procedure

1. From the menu, select **Configure > Cluster Settings**.
2. From the Cluster Settings pane, select **Security**.
3. On the Security page, select **Login Banner**.



4. Select **Edit Banner**.



5. In the Edit Login Banner, write your organization statement in the banner text box.

6. Select **Save**.
7. To prevent displaying the login banner, select **Disable Login Banner**.
8. To clear the banner text, select **Clear Login Banner Message**.

Chapter 18: User management

The management of users licensed to work with the Content Software for File system is described.

Types of users

Access to a Content Software for File system cluster is controlled by creating, modifying and deleting users. Up to 128 local users can be defined to work with a system cluster. Each user is identified by a username and must provide a password for authentication to work with the Content Software for File system GUI or CLI.

Every Content Software for File system user has one of the following defined roles:

- Cluster Admin: A user with additional privileges, as described in Cluster admin role privileges (on page 220).
- Organization Admin: A user with additional privileges within an organization (when working with different organizations, as described in Organization admin role privileges (on page 234)).
- Read-only: A user with read-only privileges.
- Regular: A user that is only used for mounting filesystems. This user can sign in to obtain an access token and change the password but cannot access the GUI or run other CLI/API commands.

Cluster Admin (the first user)

By default, when a Content Software for File cluster is created, a first user with an admin username and password is created. This user has a Cluster Admin role, which allows running all commands.

Cluster Admin users are responsible for managing the cluster as a whole. When using multiple organizations, there is a difference between managing a single organization and managing the cluster because managing the cluster also covers the management of the cluster hardware and resources. These are the additional permissions given to a Cluster Admin compared to an Organization Admin.

A Content Software for File system cluster must have at least one defined internal Cluster Admin user. However, it is possible to create a Cluster Admin user with a different name and delete the default admin user, if required.

Cluster admin role privileges

Cluster Admin users have additional privileges over regular users. These include the ability to:

- Create new users.
- Delete existing users.
- Change user passwords.
- Set user roles.
- Manage LDAP configurations.
- Manage organizations.

Additionally, the following restrictions are implemented for Cluster Admin users, to avoid situations where a Cluster Admin loses access to a Content Software for File system cluster:

- Cluster Admins cannot delete themselves.
- Cluster Admins cannot change their role to a regular user role.

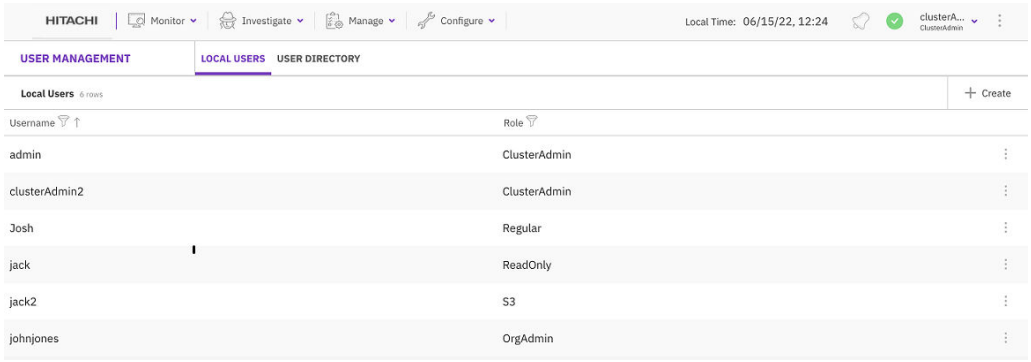
Managing users using the GUI

Using the GUI, you can:

- Manage local user
- Manage the user directory

Manage local users

Local users are created in the local system as opposed to domain users that are managed by the organization's User Directory. You can create up to 1152 local users to work with a Content Software for File system cluster.

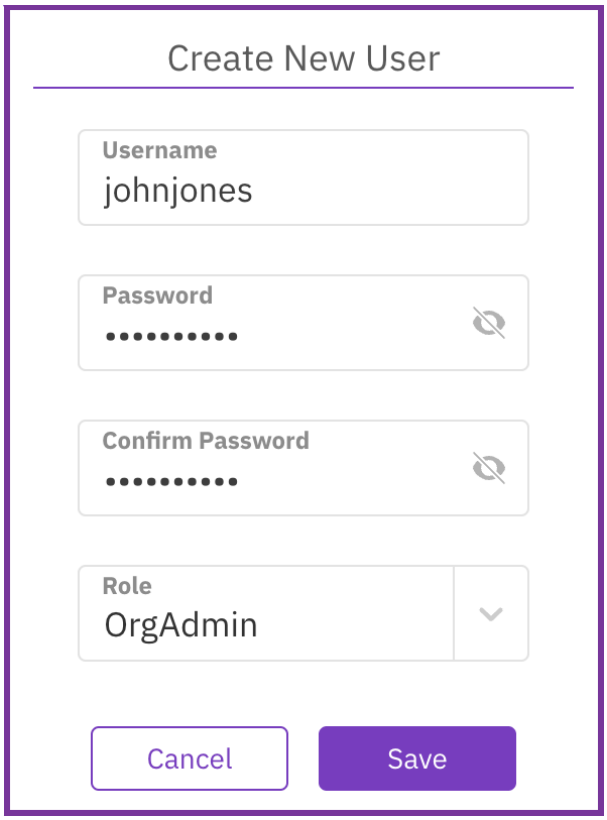


Creating a local user

Procedure

1. From the menu, select **Configure > User Management**.

2. In the Local Users tab, select **+Create**.
3. In the Create New User dialog, set the following properties:
   - **Username**: Set the user name for the local user.
   - **Password**: Set a password according to the requirements. The password must contain at least 8 characters, an uppercase letter, a lowercase letter, and a number or a special character.
   - **Confirm Password**: Type the same password again.
   - **Role**: Select the role for the local user.
4. Select **Save**.

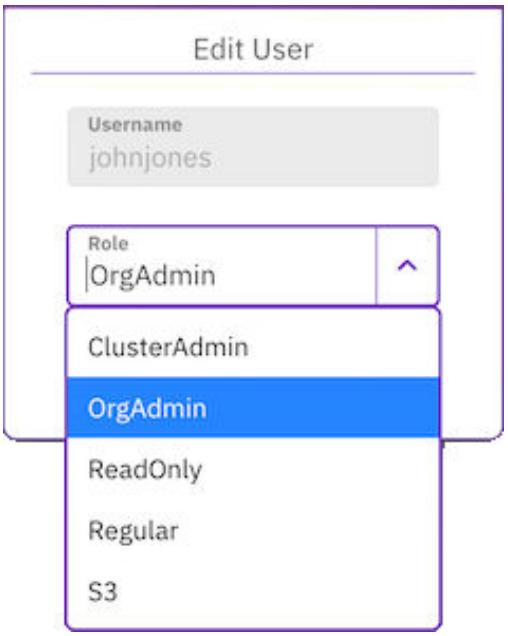


Editing a local user

You can modify the role of a local user, but not the role of an S3 user or your own role (the signed-in user).

Procedure

1. In the Local Users tab, select the three dots of the local user you want to edit, then select **Edit User**.
2. From the Role property, select the required role.
3. Select **Save**.

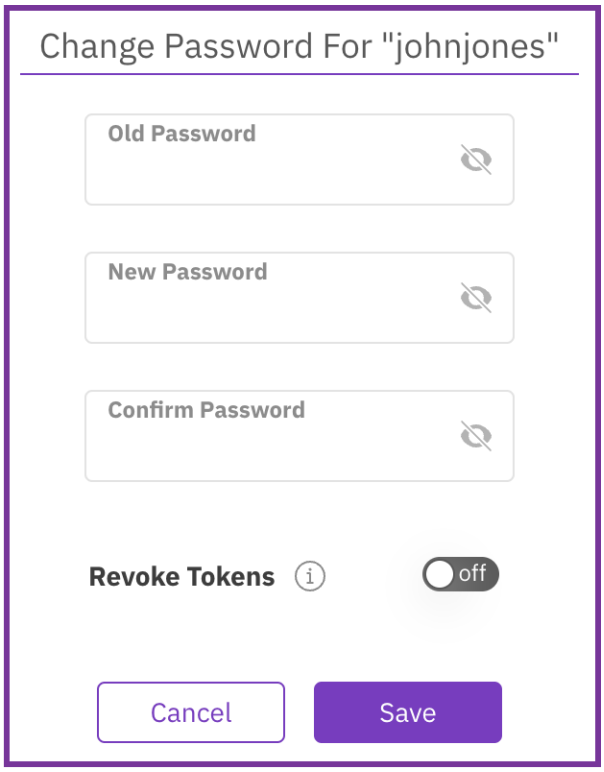


Changing a local user password

As a Cluster Admin or Organization Admin, you can change the password of a local user and revoke the user's tokens.

Procedure

1. In the Local Users tab, select the three dots of the local user you want to change the password for, then select **Change Password**.
2. In the Change Password for a user dialog, set the following properties:
   - **Old password**: Set the old password.
   - **Password**: Set a new password according to the requirements.
   - **Confirm Password**: Type the same new password again.
   - **Revoke Tokens**: If the user's existing tokens are compromised, you can revoke all the user's tokens along with changing the user's password. To re-access the system, the user re-authenticates with the new password, or the user needs to obtain new tokens using the API.
3. Select **Save**.

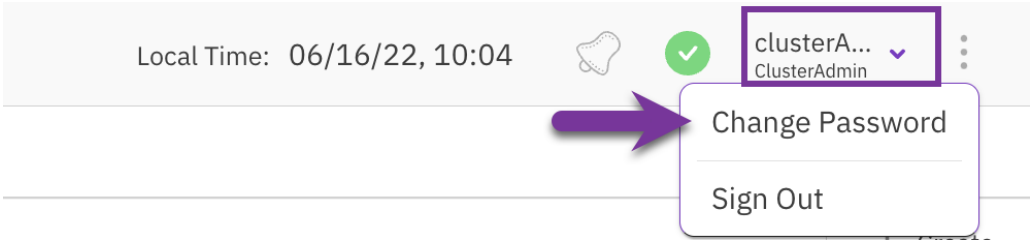


Changing your own password

You can change your own password at any time.

Procedure

1. From the top bar, select the signed-in user, then select **Change Password**.



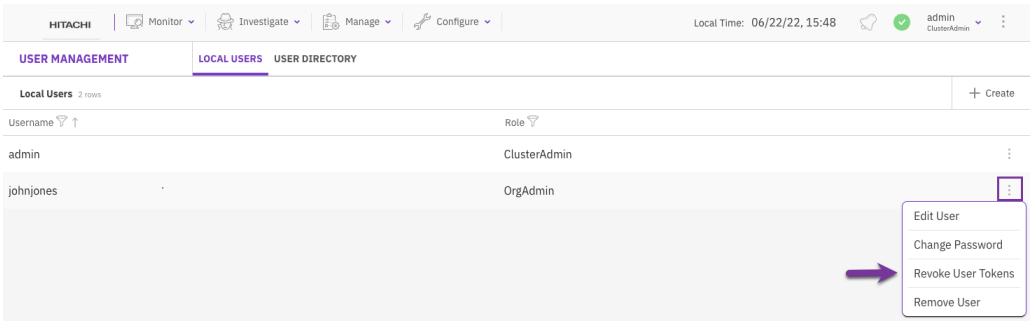
2. In the Change Password dialog set the properties as described in the Changing a local user password (on page 222) topic,
3. Select **Save**.

Revoking local user tokens

If the user's existing tokens are compromised, you can revoke all the user's tokens, regardless of changing the user's password. To re-access the system, the user re-authenticates with the new password, or the user needs to obtain new tokens using the API.

Procedure

1. In the Local Users tab, select the three dots of the local user you want to revoke the user tokens, then select **Revoke User Tokens**.



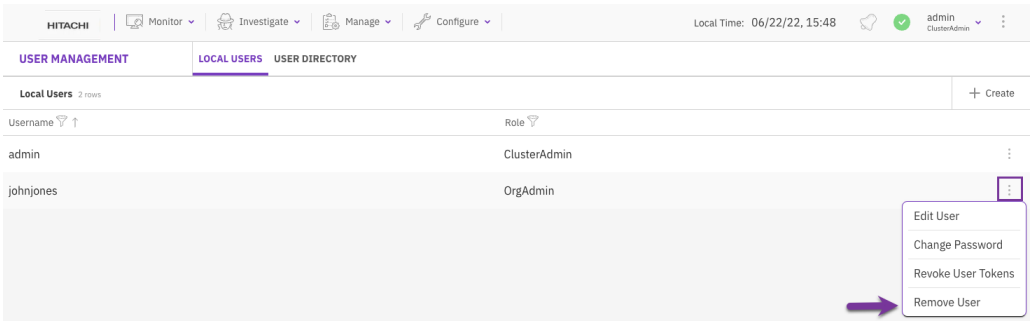
2. In the confirmation message, select **Revoke Tokens**.

Remove a local user

You can remove a local user that is no longer required.

Procedure

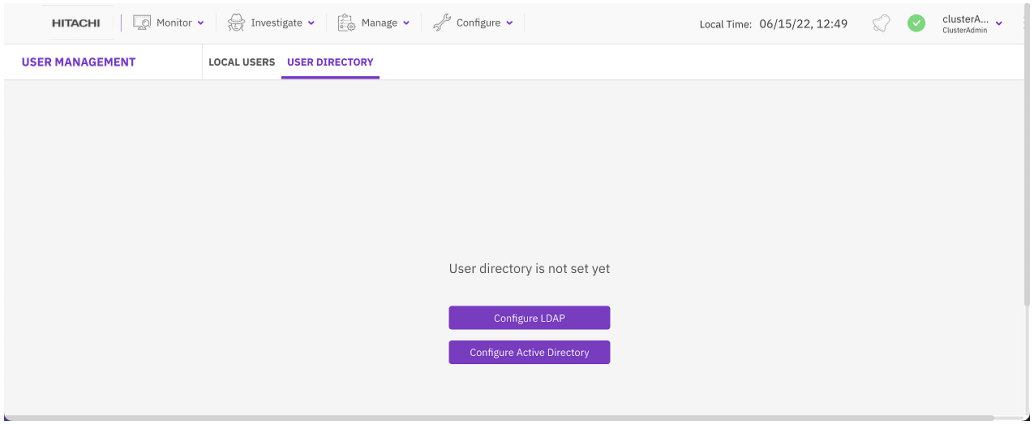
1. In the Local Users tab, select the three dots of the local user to remove, then select **Remove User**.



2. In the confirmation message, select **Yes**.

Managing user directory

You can set user access to the Content Software for File system from the organization user directory, either by LDAP directory or Active Directory.

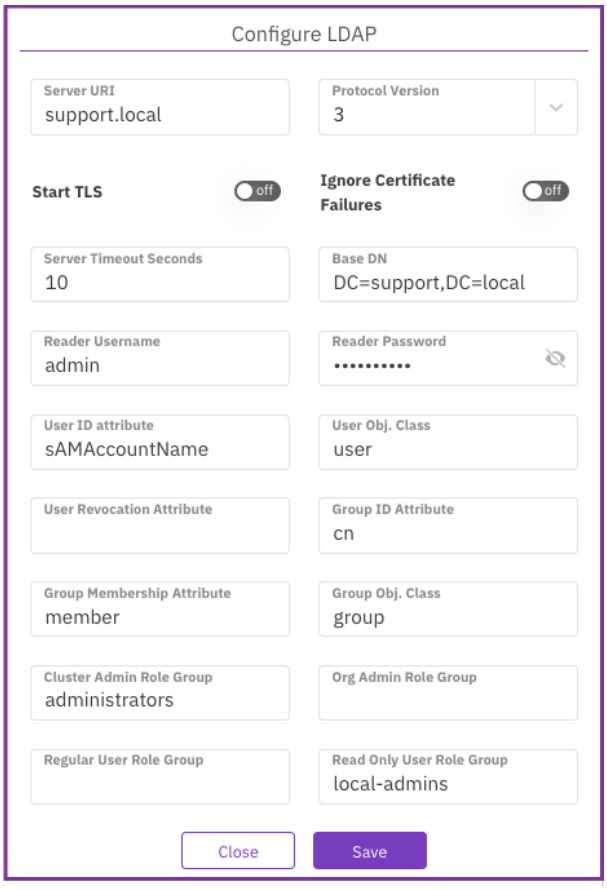


Configuring LDAP

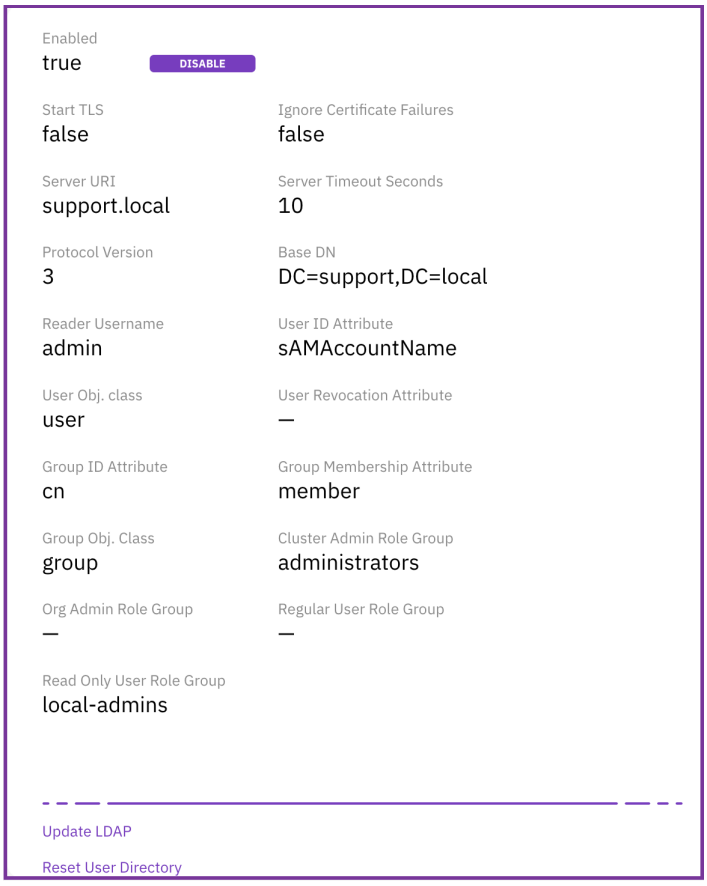
To use LDAP directory for authenticating users, you need to configure the corresponding values in the LDAP Configuration dialog.

Procedure

1. From the menu, select **Configure > User Management**.
2. Select the User Directory tab.
3. Select **Configure LDAP**.
4. Set all properties according to the organization's LDAP details.
5. Select **Save**.



Once the LDAP configuration completes, the User Directory tab displays the details. You can disable the LDAP configuration, update the configuration, or reset the configuration values.

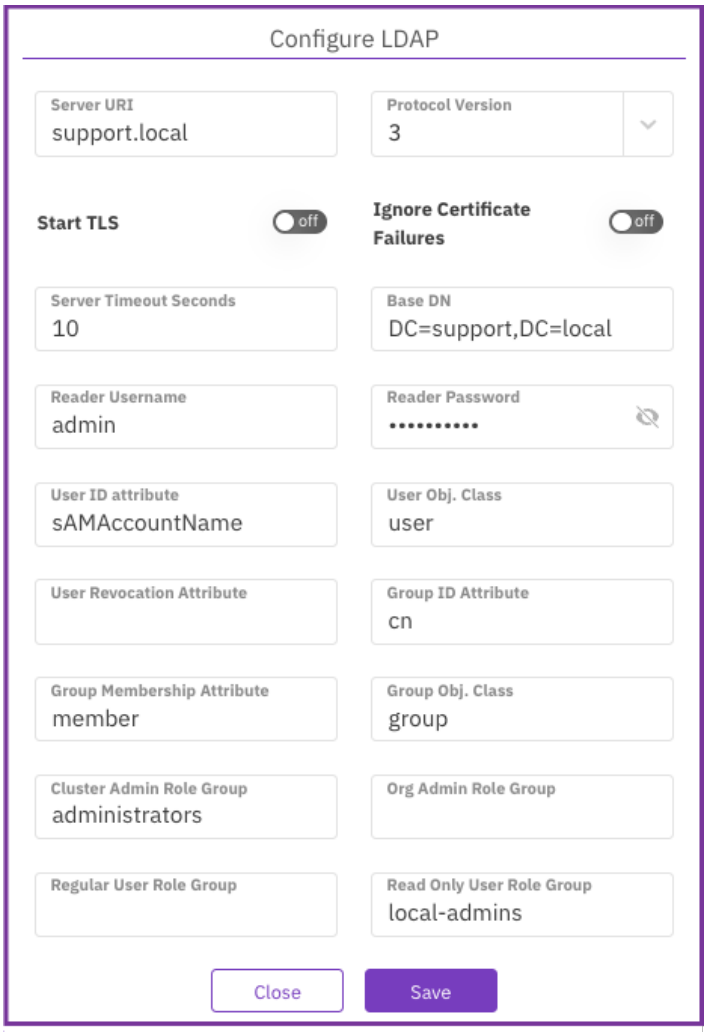


Configuring Active Directory

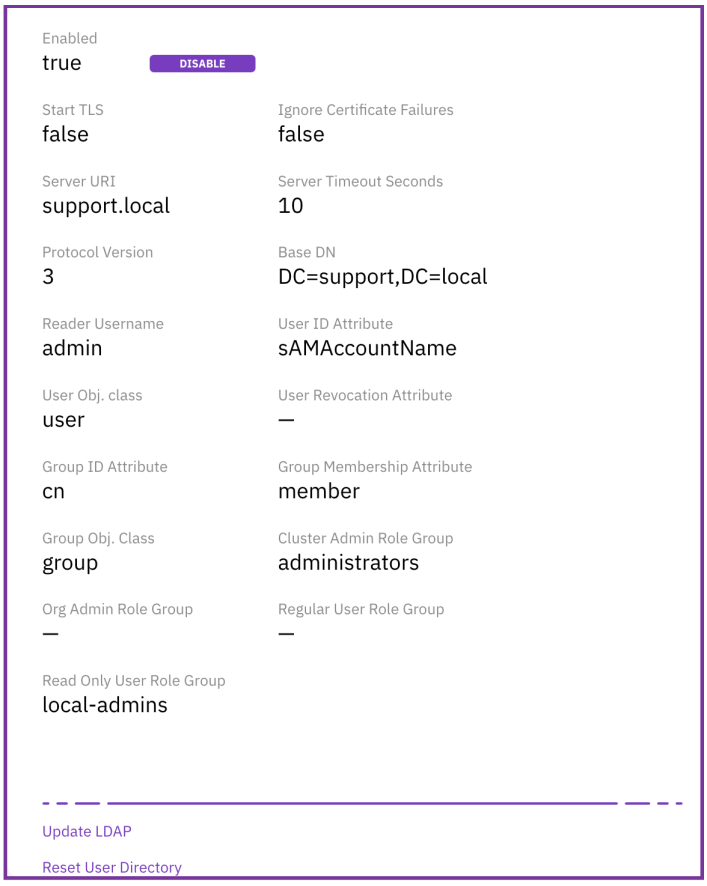
To use Active Directory for authenticating users, you configure the corresponding values in the Active Directory Configuration dialog.

Procedure

1. From the menu, select **Configure > User Management**.
2. Select the User Directory tab.
3. Select **Configure Active Directory**.
4. Set all properties according to the organization's Active Directory details.
5. Select **Save**.



Once the Active Directory configuration completes, the User Directory tab displays the details. You can disable the Active Directory configuration, update the configuration, or reset the configuration values.



Managing users using the CLI

How to manage users using the CLI

Creating users

Command

`weka user add`

Use the following command line to create a user:

```
weka user add <username> <role> [password]
```

For example:

```
$ weka user add my_new_user S3cret regular
```

This command line creates a user with a username of `my_new_user`, a password of `S3cret` and a role of `Regular` user. It is then possible to display a list of users and verify that the user was created:

```
$ weka user
Username    | Source   | Role
------------+----------+--------
my_new_user | Internal | Regular
admin       | Internal | Admin
```

Using the **`weka user whoami`** command, it is possible to receive information about the current user running the command.

To use the new user credentials, use the *WEKA_USERNAME* and *WEKA_PASSWORD* environment variables:

```
Username    | Source   | Role
------------+----------+--------
my_new_user | Internal | Regular
```

To view the parameters for the **`weka user add`** command, see the *Content Software for File Command Line Reference Guide*.

Changing user password

Command

`weka user passwd`

Use the following command line to change a local user password:

```
weka user passwd <password> [--username username]
```

Note: If necessary, provide or set *WEKA_USERNAME* or *WEKA_PASSWORD*.

To view the **`weka user passwd`** parameters, see the *Content Software for File Command Line Reference Guide*.

Deleting users

Command: **`weka user delete`**

To delete a user, use the following command line:

```
weka user delete <username>
```

For example:

```
$ weka user add my_new_user
```

Then run the **weka user** command to verify that the user was deleted:

```
$ weka user
Username | Source   | Role
---------+----------+------
admin    | Internal | Admin
```

To view the parameters for the **weka user delete** command, see the *Content Software for File Command Line Reference Guide.*

User log in

When a login is attempted, the user is first searched in the list of internal users, that is, users created using the **weka user add** command.

However, if a user does not exist in the Content Software for File system but does exist in an LDAP directory, it is possible to configure the LDAP user directory to the Content Software for File system. This will enable a search for the user in the directory, followed by password verification.

- On each successful login, a UserLoggedIn event is issued, containing the username, role and whether the user is an internal or LDAP user.
- When a login fails, an `Invalid username or password` message is displayed and a UserLoginFailed event is issued, containing the username and the reason for the login failure.

When users open the GUI, they are prompted to provide their username and password. To pass username and password to the CLI, use the *WEKA_USERNAME* and *WEKA_PASSWORD* environment variables.

Alternatively, it is possible to log into the CLI as a specific user using the **weka user login <username> <password>** command. This will run each CLI command from that user. When a user logs in, a token file is created to be used for authentication (default to `~/.weka/auth-token.json`, which can be changed using the `--path` attribute). To see the logged-in CLI user, run the **weka user whoami** command.

Note: The **weka user login** command is persistent, but only applies to the host on which it was set.

Note: If the *WEKA_USERNAME/WEKA_PASSWORD* environment variables are not specified, the CLI uses the default token file. If no CLI user is explicitly logged-in, and no token file is present, the CLI uses the default admin/admin.

To use a non-default path for the token file, use the `WEKA_TOKEN` environment variable.

For additional details on first user log in, see Cluster Admin (the first user) (on page 219).

Chapter 19: Organizations management

Organizations are used for the separation of duties between different groups of users on the same Content Software for File system. So that an organization cannot control or view other organization data. It is possible to create up to 64 organizations.

Within an organization, the Organization Admin manages the logical entities participating in obtaining control of data (the Cluster Admin cannot manage these entities).

The Cluster Admin can perform the following activities:

- Create new organizations and define the Organization Admin.
- Delete existing organizations.
- Monitor per organization the total capacity used by all the organization filesystems.

While Cluster Admins are people trusted by the different organizations (for example, have root access to the backend hosts), they are obscured from the organization data in the Content Software for File system. The Cluster Admin separation is partial, for example, they can still see the events of all organizations. The Content Software for File system ensures the separation of any sensitive information between the different organizations.

Note: The data at the hardware level is not separated. While the Content Software for File system is highly scalable and serves IOs fairly among filesystems, there is no QoS guarantee between organizations. The system limits are according to the entire system. Consequently, a single organization's workload or configuration can exhaust the entire cluster limits.

Organization management use cases

Private cloud multi-tenancy

Working with organizations makes it possible to manage different departments. While this requires more configuration, for example, different LDAP configurations are usually unnecessary between different departments in the same organization, the Cluster Admin is fully trusted.

It is possible to separate and obscure specific departments, such as IT, Finance, Life Sciences, Genomics, and even specific projects in departments.

Logical separation of external groups of users

When multiple, independent groups use the same provided infrastructure, the use of multiple organizations provides much better security, obscuration, and separation of data.

Cluster level entities

The Cluster Admin manages the following entities at the cluster level:

- Hardware.
- NFS service (NFS groups and IP/interfaces)
- SMB service.
- Filesystem groups - definition of tiering policies for the different groups, while the Organization Admin selects the filesystem group from the predefined list of groups for each filesystem created
- KMS.

Organization level entities

At the organization level, only the relevant Organization Admin manages all system entities, while the users can only view the system entities within the organization.

Cluster Admins do not have permissions to view or manage the system entities within the organization, which include the following:

- Filesystems, and the option to mount the filesystems (also a Cluster Adminfile cannot mount the filesystems)
- Object store buckets.
- LDAP server.
- NFS exports (NFS client permissions).

Different protocols are not supported other than in the root organization.

Note: Different protocols are not supported other than in the root organization.

Note: Only exports of the 'legacy' NFS stack can be managed within a non-root organization.

Managing organizations

Only users defined as Cluster Admins can manage organizations. When no organization is created, the root organization is the default organization and all operations are regular. That is, it is not necessary to authenticate the mounts or supply an organization name when logging in using the GUI/CLI.

Once a new organization is created, the organization name must be provided in every login command, using the `--org` attribute in the **`weka user login`** command.

Usage and quota management

Cluster Admins can view an organization's usage (both SSD and total) and can limit usage with quotas per organization. This can be leveraged for charge-backs on either used or allocated capacity of SSD or object store data.

Organization admin role privileges

When a new organization is created, the Cluster Admin creates an Organization Admin user for the organization, who is the administrator within the organization responsible for managing each organization level entity (on page 233).

Organization Admins have similar privileges to Cluster Admins, except that these privileges are limited to the organization level. They can perform the following within the organization:

- Create new users.
- Delete existing users.
- Change user passwords.
- Set user roles.
- Manage the organization LDAP configuration.

To avoid situations where an Organization Admin loses access to a Content Software for File system cluster, the following restrictions are implemented on Organization Admins:

- Cannot delete themselves.
- Cannot change their role

Managing organizations using the GUI

Using the GUI, you can:

- Create an organization
- View organizations
- Edit an organization
- Delete an organization

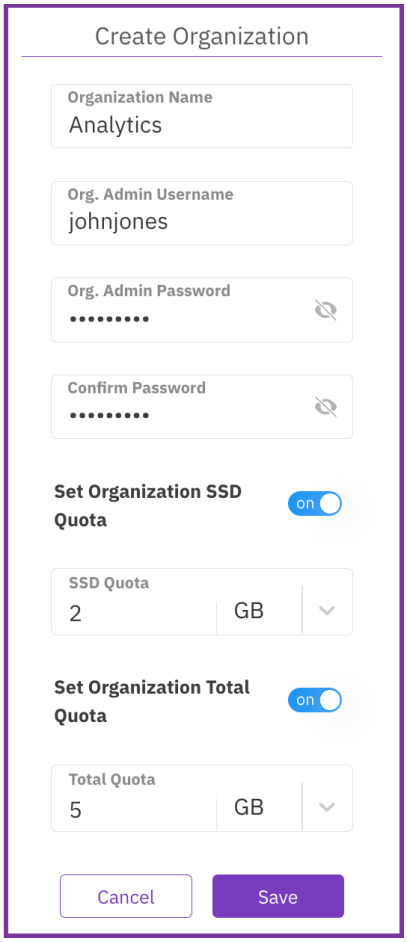
Creating an organization using the GUI

Only a Cluster Admin can create an organization.

Procedure

1. From the menu, select **Configure > Organizations**.
2. On the Organizations page, select **+Create**.

3. In the Create Organization dialog, set the following properties:
   - **Organization Name**: A name for the organization.
   - **Org. Admin Username**: The user with an Organization Admin role created for the organization.
   - **Org. Admin Password**: The password of the user with an Organization Admin role created for the organization.
   - **Confirm Password**: The same password as set in the Org. Admin Password.
   - **Set Organization SSD Quota**: Turn on the switch and set the SSD capacity limitation for the organization.
   - **Set Organization Total Quota**: Turn on the switch and set the total capacity limitation for the organization (SSD and object store bucket).
4. Select **Save**.



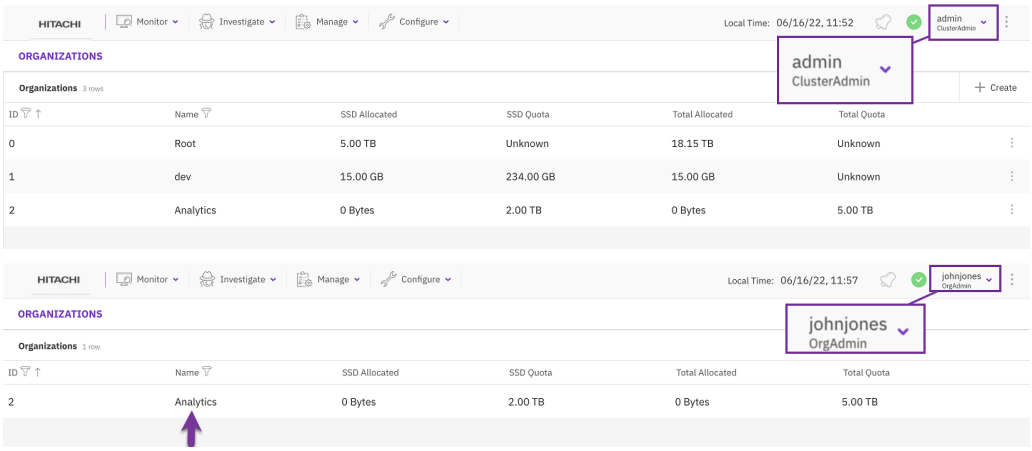
Viewing organizations

As a Cluster Admin, you can view all organizations in the cluster.

As an Organization Admin, you can view only the organization you are assigned to.

Procedure

1. From the menu, select **Configure > Organizations**.

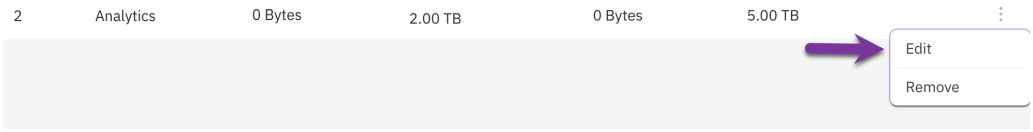


Editing an organization

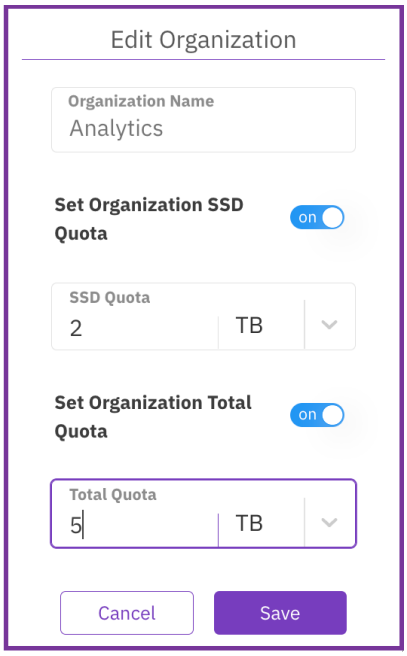
You can modify an organization's SSD and total quota to meet the capacity demand changes.

Procedure

1. From the menu, select **Configure > Organizations**.
2. On the Organizations tab, select the three dots of the organization to edit and select **Edit**.



3. In the Edit Organization dialog, set the following properties:
   - **Set Organization SSD Quota**: Turn on the switch and set the SSD capacity limitation for the organization.
   - **Set Organization Total Quota**: Turn on the switch and set the total capacity limitation for the organization (SSD and object store bucket).



4. Select **Save**.

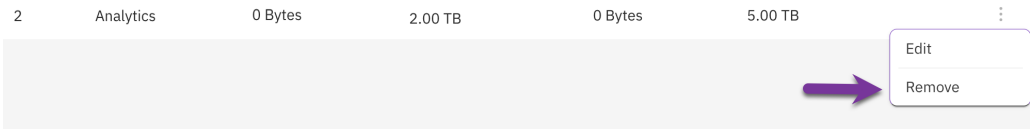
Deleting an organization

If an organization is no longer required, you can remove it. You cannot remove the root organization.

Note: Deleting an organization is irreversible. It removes all entities related to the organization, such as filesystems, object stores, and users.

Procedure

1. From the menu, select **Configure > Organizations**.
2. On the Organizations tab, select the three dots of the organization to edit and select **Remove**.



3. In the confirmation message, select **Yes**.

Mount authentication for organization filesystems

Once the Cluster Admin has created an organization and the Organization Admin has created filesystems, users, or configured the LDAP for the organization, regular users of the organization can mount filesystems.

The purpose of organizations is to provide separation and security for organization data, which requires authentication of the Content Software for File system filesystem mounts. This authentication of mounts prevents users of other organizations and even the Cluster Admin from accessing organization filesystems.

Mounting filesystems in an organization (other than the Root organization) is only supported using a stateless client. If the user is not logged into the Content Software for File system, a login prompt will appear as part of the mount command.

Mounting a filesystem using the CLI

To securely mount a filesystem, first log into the Content Software for File system:

```
weka user login my_user my_password --org my_org -H backend-host-0
```

Then mount the filesystem:

```
mount -t wekafs backend-host-0/my_fs /mnt/weka/my_fs
```

Mount authentication

Authentication is achieved by obtaining a mount token and including it in the mount command. Logging into the Content Software for File system using the CLI (the `weka user login` command) creates an authentication token and saves it in the client (default to `~/.weka/auth-token.json`, which can be changed using the`--path`attribute).

The Content Software for File system assigns the token that relates to a specific organization. Only mounts that pass the path to a correct token can successfully access the filesystems of the organization.

Once the system authenticates a user, the mount command uses the default location of the authentication token. It is possible to change the token location/name and pass it as a parameter in the mount command using the `auth_token_path` mount option, or the`WEKA_TOKEN` environment variable.

```
mount -t wekafs backend-host-0/my_fs /mnt/weka/my_fs -o auth_token_path=<path>
```

This option is useful when mounting several filesystems for several users/organizations on the same host or when using Autofs.

When a token is compromised or no longer required, such as when a user leaves the organization, the Organization Admin can prevent using that token for new mounts by revoking the user access.

Chapter 20: Expanding and shrinking cluster resources

How to expand and shrink a cluster in a homogeneous Content Software for File system configuration.

Note: The cluster expansion process described here is only applicable to a homogeneous Content Software for File system configuration, which is highly recommended. For non-homogeneous system configurations, contact your Hitachi representative.

Expand and shrink overview

An overview of the cluster expand and shrink process in a homogeneous Content Software for File system configuration is provided.

In the Content Software for File system, it is possible to expand and shrink a cluster as follows:

- Add or delete backend hosts
- Add or delete SSDs from an existing backend host
- Change the number of cores assigned to the Content Software for File system in existing backend hosts
- Change the amount of memory allocated to the Content Software for File system in existing backend hosts
- Change the network resources assigned to the Content Software for File system in existing backend hosts

Note: The expansion or shrinking of networking resources is performed infrequently.

Note: The cluster expansion process described here is only applicable to a homogeneous Content Software for File system configuration, which is highly recommended.

Planning an expansion or shrink

Note: The expansion of a Content Software for File system offers the opportunity to increase performance, while the shrinking of a system may reduce performance. For more details and to receive estimates contact your Hitachi representative.

Note: In the following descriptions, cluster expansion also relates to cluster shrinking.

Expansion procedures are similar to installation instructions and can be obtained as a Content Software for File system installation procedure, available if you contact customer support. Similar to planning a new cluster, the objectives of the expansion, in terms of space and performance, need to be translated to the actual cluster resources. This process is practically a repeat of the planning process for new clusters, with the following options and limitations provided in the next sections.

Possible expansion options

- Addition of new backend hosts.
- Addition of new failure domains, as long the system was installed with failure domains.
- Addition of new SSDs to existing backend hosts.
- Assignment of additional cores to Content Software for File in existing backend hosts.
- Assignment of more memory to Content Software for File in existing backend hosts.
- Assignment of additional network resources to Content Software for File in existing backend hosts.
- Reconfiguration of hot spares.

Expansion limitations

- It is not possible to change the defined Content Software for File system protection scheme.
- It is not possible to define failure domains on a system that was installed without failure domains.
- A Content Software for File system configured with failure domains cannot be configured to be without failure domains.
- Only the same network technology can be implemented that is, it is not possible to mix between Ethernet and InfiniBand.

To plan the capacity of the Content Software for File system after expansion, refer to SSD capacity management (on page 21).

Cluster expansion process

Once an expansion of more SSDs or backend hosts has been planned and executed, the Content Software for File system starts a redistribution process. This involves the redistribution of all the existing data to be perfectly balanced between the original hosts or SSDs and newly added resources. This process can take from minutes to hours, depending on the capacity and the networking CPU resources. However, the capacity increase is instant, and therefore it is possible to define more filesystems immediately, without waiting for the completion of the redistribution process.

Note: If necessary, contact customer support for more details on the redistribution process and its expected duration.

Once the expansion of more cores or backend hosts has been implemented, the added CPU resources are operational in less than a minute. Write performance improves almost immediately, while read performance only improves on completion of the redistribution of the data.

Note: As part of the requirements for a homogeneous Content Software for File system configuration, when expanding memory resources, the new hosts must have the same memory as the existing hosts.

Hitachi Vantara

Corporate Headquarters
2535 Augustine Drive
Santa Clara, CA 95054 USA

HitachiVantara.com/contact