CLOUDIAN®

Cloudian HyperStore® Technical Guide

Table of Contents

INTRODUCTION

With the popularity of rich media, the proliferation of mobile devices and the digitization of content, there has been and continues to be exponential growth in the quantity of unstructured data that IT is managing. In fact, IDC predicts that all data will grow to 40 zettabytes by 2020, resulting in a 50-fold growth from the beginning of 2010. 90% of this data growth will be unstructured. The growth is not slowing down; in fact, it continues to accelerate in size and scope.

This explosive growth in data and content is simply not sustainable for current NAS and SAN infrastructures. Backups and restores are taking longer. Migrating data from older storage systems to new storage systems are labor intensive and expensive. Provisioning storage for users is more frequent and time consuming.

Not only does all this unstructured data increase the cost of managing the infrastructure, it also impacts the internal consumers of storage. Most IT organizations are faced with a flat to declining storage budget and are forced to manage the ever-increasing storage with the same or reduced IT resources. In short, the costs and complexity of traditional storage systems continue to increase. A radical change in storage infrastructure is needed if enterprise IT is ever going to tame the data explosion.

A NEW TYPE OF STORAGE

Software-defined object storage offers an alternative approach to NAS/SAN systems based on expensive proprietary hardware. It gives enterprises the ability to leverage the latest advancements in cost-effective commodity CPU and storage technology. With object storage, enterprise environments can keep up with the massive storage growth and IO demands of critical business applications. Software-defined storage (SDS) architectures along with more powerful CPUs lead to greater scale and performance. Intel, for example, typically comes out with a new CPU product every 18-24 months. Compare this with the typical 3-year product refresh cycle from proprietary storage vendors. It is clear why IT organizations need more flexibility. In addition, manufacturers continue to drive innovation into the hard disk drive market space, delivering increased disk drive densities and a lower cost per gigabyte (GB). This new architecture allows enterprises to take advantage of these technology updates earlier, to meet the explosive storage growth demand and gain cost efficiencies in the data center.

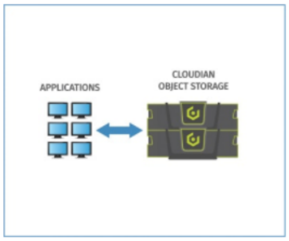
Cloudian HyperStore® is a scale-out object storage system designed to manage massive amounts of unstructured data. It is an SDS platform which runs on any standard x64 server platform. This dramatically reduces the cost for datacenter storage while still providing limitless scalability, extreme availability and unprecedented reliability.

In this whitepaper, we will provide an in-depth view of Cloudian HyperStore by providing insight into the overall system architecture. A deep-dive into the internal components from a technical implementation perspective will help you design and deploy HyperStore. Throughout this document the unique capabilities of the product will be highlighted and expounded upon.

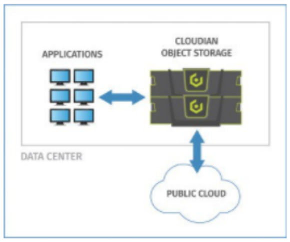
CLOUDIAN HYPERSTORE OVERVIEW

Cloudian HyperStore enables data centers to provide highly cost-effective on-premise unstructured data storage repositories. Cloudian HyperStore is built on standard hardware that spans across the enterprise as well as out into public cloud environments.

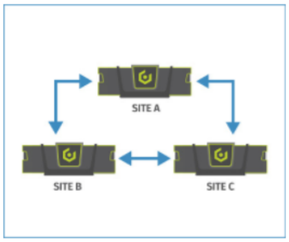
Cloudian HyperStore is available as a stand-alone software or fully integrated with hardware as a Cloudian HyperStore appliance. It easily scales to limitless capacities and offers multi-datacenter storage. HyperStore also has fully automated data tiering to all major public clouds, including AWS, Azure and Google Cloud Platform. It fully supports S3 applications and has flexible security options. HyperStore deployment models include on-premises storage, distributed storage, storage-as-a-service or even other combinations as illustrated below.



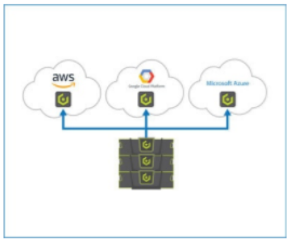
ON-PREMISES



HYBRID CLOUD



MULTI-SITE



MULTI-CLOUD

CLOUDIAN HYPERSTORE FEATURES

Cloudian Hyperstore software can be deployed on existing hardware or pre-installed on Cloudian HyperStore appliances. offers robust availability with system management control, monitoring capabilities and reporting. Cloudian HyperStore boasts a host of features including enterprise NAS when used with HyperFile.

Cloudian HyperStore offers multi-tenancy, WORM, hybrid cloud streaming and configurable storage policies with flexible protection levels and redundancy through ISA-L erasure coding, replication factors, data compression and server-side encryption. With Cloudian HyperStore, seamless data management is possible allowing users on demand access to their data anywhere and anytime. Built on a robust object storage platform for effortless data sharing, cloud service providers around the world use Cloudian HyperStore to deploy and manage both public and private clouds, while enterprises rely on it for private and hybrid clouds.

EASY TO INSTALL

Cloudian includes automated installer tools and wizards, allowing a three-node cluster of appliances to be installed in as little as two hours. A knowledge of Linux server management and network configuration is sufficient for a basic install.

LIMITLESSLY SCALES ON DEMAND

Cloudian HyperStore offers flexible growth and infinite scalability options. No matter your storage demand, you can seamlessly grow your storage in any dimension as fast or as slow as desired. You can add heterogeneous datacenters and regions and/or add a single node or multiple nodes, all in one operation. You can even add a different number of nodes in each of your datacenters if your storage demand requires it. For example, you could add five nodes in DC1, eight in DC2 and one in DC3.

ENTERPRISE DATA PROTECTION

Cloudian HyperStore offers true enterprise data protection. With replication factors and the ISA-L Erasure Coding, Cloudian HyperStore optimizes storage protection for all data objects. Data protection and durably choices are flexible, enabling efficient storage redundancy to meet your specific business needs.

EFFORTLESS DATA MOVEMENT

Not only does Cloudian HyperStore scale on demand, it simplifies data management. Cloudian HyperStore enables storing and retrieving your data where you want, when you want, using unique features like object streaming and dynamic auto-tiering. Data can move seamlessly between your on-premises storage and the public cloud regardless of data type and size.

S3 COMPATIBLE

With Amazon setting the cloud storage standard making it the largest object storage environment, and Amazon S3 API becoming the de facto standard for developers writing storage applications for cloud, it is imperative that every cloud, hybrid storage solution be S3 compliant. Cloudian HyperStore, in addition to being S3 compliant, also offers the flexibility to be on-prem object storage and integrate as a hybrid tier to public cloud providers such as AWS, Microsoft Azure and Google Cloud.

SECURITY

Cloudian HyperStore takes safeguarding your data very seriously. Two server-side encryption methods (SSE/SSE-c, Keysecure) are implemented to ensure that the data is always protected. HyperStore also supports the option of using a third-party Key Management System to generate and manage the encryption key (KMS). This relieves administrators from the burden of encryption key management and eliminates the risk of data loss occurring due to lost keys. Encryption can be managed very granularly—either at a bucket level or down to an individual object.

MULTI-TENANCY

Advanced identity and access managed features allow system administrators to provision and manage groups and users, define specific classes of service for groups and users and configure billing and charge-back policies. Multiple credentials per user are also supported. Configurable group and user level QoS rate limits ensure groups and users do not exceed storage quotas and allows for multi access in a way that bandwidth is not throttled affecting other tenants.

IAM USER SUPPORT

HyperStore provides selective support for the Amazon Identity and Access Management (IAM) API. This support enables each HyperStore user, under his or her HyperStore user account, to create IAM groups and IAM users. As with Amazon, all S3 object data created by IAM users belongs to the parent HyperStore user account, (otherwise known as the “root” account). IAM users can be deleted by their HyperStore parent user without any S3 object data being deleted.

INTEGRATED BILLING, MANAGEMENT AND MONITORING

The HyperStore system maintains comprehensive service usage data for each group and each user in the system. This usage data, which is protected by replication, serves as the foundation for HyperStore service billing functionality. The system allows for creation of rating plans that categorizes types of service usage for a singles users or groups for a selected service period. The CMC (Cloudian Management Console) has a function for displaying a single user’s bill report in a browser, HyperStore Admin API can be used to generate user or group billing data that can be ingested a third-party billing application. Cloudian HyperStore also allows for special treatment of designated source IP addresses, so that the billing mechanism does not apply any data transfer charges for data coming from or going to these “whitelisted” domains.

BROAD APPLICATION SUPPORT

With complete S3 compatibility, Cloudian HyperStore ensures seamless S3 integration with every available AWS/ S3 application. Cloudian HyperStore allows unmatched customer choice in deploying applications and storage on-and off-premises. The highly active S3 developer community generates lots of innovative applications in categories including: enterprise secure file sharing; backup, data retention and archiving; NFS/CIFS gateways; and desktop file storage and backup; Cloudian HyperStore uniquely supports them all.

WORM

HyperStore supports applying a Write Once, Read Many (WORM) policy to a bucket via an advanced S3 extension. When a WORM policy is implemented for a bucket, objects in the bucket cannot be altered or deleted through HyperStore S3 interfaces until the object age exceeds a specified retention period. This type of policy can be used, to meet regulatory requirements that call for data to be kept in its original form throughout a mandated retention period. WORM is implemented on a per-bucket basis and therefore offers great multi-tenancy retention flexibility.

LIMITLESS SCALIBILITY ARCHITECTURE

Built on Cassandra no-SQL database, Cloudian HyperStore can store vast amounts of unstructured data without object size limitations. This gives Cloudian HyperStore improved storage scaling control over data availability.

HOW WE DO IT

Cloudian HyperStore solution is built on open scalability of S3 and Cassandra, an architecture that originated at large scale cloud companies like Google, Facebook and Amazon.

Intelligence in Software

100% software-defined All data, metadata, with no reliance on any special hardware for PB-scale durability, availability, and storage.

Distributed Everything

All data, metadata operations configurations, and Operations are distributed across the cluster for scale-out.

Extreme Durability

Designed to tolerate disk, node, rack datacenter failure, and detect bit-rot and network errors.

Multi-Tenant Architecture

Designed from scratch to isolate and protect tenant data with built-in QoS, billing, and reporting.

DISTRIBUTED PEER-TO-PEER OBJECT STORAGE

Cloudian storage clouds are implemented by deploying individual nodes comprised of CPUs and disk drives into a logical Cassandra peer-to-peer ring architecture. As physical nodes are added, all the resources are aggregated into a common pool of storage and CPU resources across the cluster. For redundancy and availability purposes, three nodes are typically deployed in an initial implementation. Single nodes can then be added as needed to scale capacity. Data is dispersed across all the nodes, via erasure coding or replication to improve availability and to enhance performance.

VIRTUAL NODES

Cloudian's vNode technology enhances data redundancy and availability a step further. The disk resources within a single node can be subdivided into smaller IO devices called vNodes. This allows for greater IO parallelism and hence greater storage IO performance across the HyperStore system. vNodes also enhance availability. With virtual nodes if a drive or a node fails, recovery processes can be distributed in parallel across all the drives within a node/appliance.

PARALLEL DISK IO DATA PROTECTION

The ability to run disk IO in parallel across multiple nodes is a critically important feature because as more devices are added, the probability that a drive will fail increases. To compound this problem, disk manufacturers now have ultra-high-density 10 terabyte (TB) disk drives which will continue to increase in capacity over time. The RAID re-build times to recover lost devices can easily take 48 hours or longer. Even RAID 6 protected storage systems, which can withstand up to two simultaneous drive failures without incurring data loss, become more vulnerable to data loss as drive rebuild times increase. By leveraging erasure coding and a scalable parallel disk IO architecture, Cloudian HyperStore dramatically shrinks drive rebuild times.

CONFIGURABLE DATA CONSISTENCY

Cloudian HyperStore also provides policy-based data consistency levels when using replication to protect objects across a cluster. For example, the default consistency requirement for read and write operations is defined as “quorum”. These means a read or write operation must succeed on a set number of replica copies before a success response is returned to the client application. This enables user flexibility in how stringent they wish their replication policy to be. For example, for those data objects that are considered mission critical, the replication policy can be set to wait until an acknowledgment is received from nodes across multiple datacenter locations before the acknowledgement is sent upstream to the application. If performance is deemed more critical, then a correspondingly fewer number of replicas may be configured within a particular quorum.

STORAGE NODE HETEROGENEITY

Cloudian’s vNode technology enables datacenters to intermix node types. In other words, storage nodes deployed into a cluster can be of different sizes. For example, a 24TB node could be installed alongside a 48TB node and the Cloudian HyperStore operating system will automatically pool and load balance these resources as they are added to the cluster. This gives flexibility in adding capacity and CPU resources to right-size their storage to meet the business needs. Efficiencies are improved as optimal resources can be added to the HyperStore object-storage cluster on demand.

COMPRESSION YOUR WAY

Cloudian HyperStore offers three different types of data compression technology—lz4, snappy, and zlib. compression can reduce storage and network consumption by up to 40 percent, while accelerating data replication speeds. With the savings from optimized data storage on disk and less data to move over the network, businesses can get more life out of their existing storage and network investments; further improving their ROI and lowering their TCO.

AUTO-TIERING

Cloudian enables seamless integration with on-premise HyperStore cloud storage and the public cloud. In particular, HyperStore supports an auto-tiering feature whereby objects can be automatically moved from local HyperStore storage to a remote storage system on a defined schedule. HyperStore supports auto-tiering from a local HyperStore bucket to any of several types of destinations systems including S3-compliant systems: Amazon S3, Amazon Glacier, Google Storage Cloud, a HyperStore region or system, or a different S3-compliant system of your choosing. In addition, HyperStore supports auto-tiering to Microsoft Azure and Spectra Logic Black Pearl. Auto-tiering is configurable on a per-bucket basis and can be enabled to happen immediately, called Bridge Mode or on a defined daily and/or weekly schedule.

QUALITY OF SERVICE (QOS)

Cloudian HyperStore provides QoS and metering tools. Storage administrators can set a maximum allowable limit on both storage consumption and IO, based on the user or a group of users, and then charge back those users on a monthly basis, just like a utility. For example, a CFO could be assigned a high priority Platinum Service Level privilege to financial records while an end-user accessing the data could be given a lower sliver service level for access. QoS and metering are foundational capabilities for implementing a multi-tenant private cloud storage solution.

CLOUDIAN HYPERSTORE ARCHITECTURE

As you've seen, Cloudian HyperStore is an Amazon S3-compliant multi-tenant object storage system with many advanced capabilities. The system utilizes a "non-SQL" (NoSQL) storage layer for maximum flexibility and scalability. The Cloudian HyperStore system enables any service provider or enterprise to deploy an S3-compliant multi-tenant storage cloud.

The Cloudian HyperStore system is designed specifically to meet the demands of high volume, multi-tenant data storage:

- **Amazon S3 API compliance.** The Cloudian HyperStore system is nearly 100% compatible with Amazon S3's HTTP REST API. Existing HTTP S3 applications will work with the Cloudian HyperStore service, and existing S3 development tools and libraries can be used for building Cloudian HyperStore client applications with full assurance of combability with the HyperStore storage system.
- **Secure multi-tenancy.** The Cloudian HyperStore system provides the capability to securely have multiple users reside on a single, shared infrastructure. Data for each user is logically separated from other users' data. This data cannot be accessed by any other user unless access permission is explicitly granted.
- **Group support.** An enterprise or work group can share a single Cloudian HyperStore account. Each group member can have dedicated storage space, and the group can be managed by a designated group administrator.
- **Quality of Service (QoS) controls.** Cloudian HyperStore system administrators can set storage quotas and usage rate limits on a per-group and per-user basis. Group administrators can set quotas and rate controls for individual members of the group.
- **Access control rights.** Read and write access controls are supported at per-bucket and per-object granularity. User data can also be securely exposed via public URLs for regular web access, subject to configurable expiration periods.
- **Reporting and billing.** The Cloudian HyperStore system supports usage reporting on a system-wide, group-wide, or individual user basis. Like a utility company, billing of groups or users can be based on storage quotas and usage rates.
- **Horizontal scalability.** Running on standard off-the-shelf hardware, a Cloudian HyperStore system can scale up to thousands of nodes across multiple datacenters, supporting millions of users and hundreds of petabytes of data. New nodes can be added without service interruption.
- **High availability.** The Cloudian HyperStore system has a fully distributed, peer-to-peer architecture, with no single point of failure. The system is resilient to network and node failures with no data loss due to the automatic replication and recovery processes inherent to the architecture. A Cloudian HyperStore geocluster can be deployed across multiple datacenters to provide redundancy and resilience in the event of a datacenter scale disaster.
- **Storage Use Monitoring**. HyperStore supports usage tracking and reporting on a system level down to a per-bucket basis. Per bucket statistics is disabled by default. Figure 1 illustrates the major components of the Cloudian HyperStore architecture: Cloudian Management Console (CMC), S3 Service, Administrative Service, Cloudian HyperStore storage services, Redis and Cassandra databases.

CLOUDIAN HYPERSTORE ARCHITECTURE OVERVIEW

The Cloudian HyperStore is a fully distributed architecture that provides no single point of failure. Flexible data protection options are available with replication or erasure coding, data recovery upon a node failure, dynamic re-balancing upon node addition, multi-datacenter and multi-region support. The illustration below shows all of the service components that comprise a Cloudian HyperStore system node.

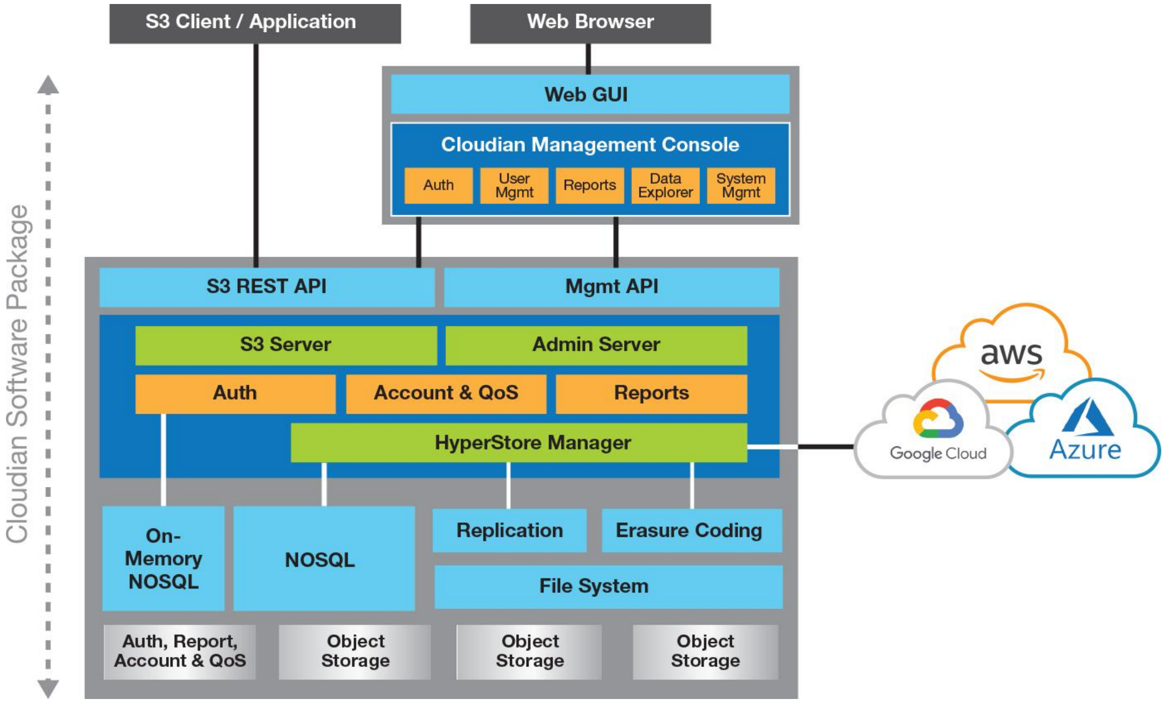


Figure 1 – Cloudian HyperStore Architectural Diagram

CLOUDIAN HYPERSTORE GEO-CLUSTER

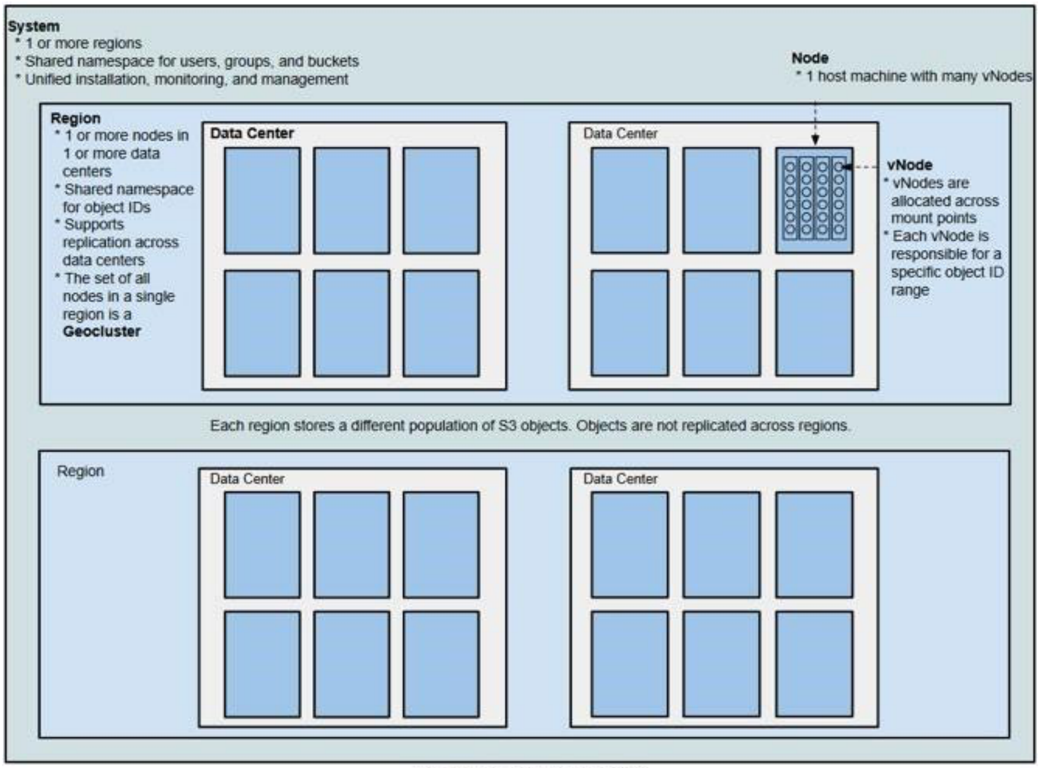
Figure 2 below shows the conceptual view on an entire Cloudian HyperStore geo-cluster that can be deployed with multiple regions, multiple datacenters, multiple nodes and multiple vNodes.

REGIONS

Like Amazon S3, the Cloudian HyperStore system supports the implementation of multiple “service regions”. Setting up the Cloudian HyperStore system to use multiple service regions is optional.

The main benefits of deploying multiple service regions are:

- Each region has its own independent Cloudian HyperStore geo-cluster for S3 object storage. Consequently, deploying multiple regions is another means of scaling out your overall Cloudian HyperStore storage system (beyond using multiple nodes and multiple datacenters to scale out a single geo-cluster). Note that in a multi-region deployment, entirely different S3 datasets are stored in each region. Each region has its own token space and there is no data replication across regions.
- With a multi-region deployment, your service users can choose the service region in which their storage buckets will be created. Users may, for example, choose to store their S3 objects in the region that’s geographically closest to them; or they may choose one region rather than another for reasons of regulatory compliance or corporate policy.


System
* 1 or more regions
* Shared namespace for users, groups, and buckets
* Unified installation, monitoring, and management
Node
* 1 host machine with many vNodes
Region
* 1 or more nodes in 1 or more data centers
* Shared namespace for object IDs
* Supports replication across data centers
* The set of all nodes in a single region is a Geocluster
Data Center
Data Center
vNode
* vNodes are allocated across mount points
* Each vNode is responsible for a specific object ID range
Each region stores a different population of S3 objects. Objects are not replicated across regions.
Region
Data Center
Data Center

HyperStore System Levels

Figure 2 – Cloudian HyperStore Geo-cluster Diagram

SERVICES

The Cloudian HyperStore system supports several types of services each of which plays a role in implementing the overall Cloudian HyperStore S3 object storage service:

SERVICE NAME	DESCRIPTION AND ROLE
S3 Service	The Cloudian HyperStore system provides a high-performant S3 proxy service. The S3 Service processes S3 REST requests incoming from client applications
Cloudian HyperStore Service and the HSFS	As an object store, Cassandra provides a wealth of valuable built-in functionality including data partitioning, automatic replication, easy cluster expansion, quorum calculations, and so on. The Cloudian HyperStore system uses a hybrid storage solution where Cassandra can optionally be used for small S3 data objects while the Linux filesystem on Cassandra nodes is used for larger S3 data objects. The area of the Linux filesystem where S3 object data is stored is called the Cloudian HyperStore File System (HSFS). In Cloudian HyperStore, Cassandra capabilities are used to determine the distributed data management information such as the nodes that a specific key should be written to and replicated. HSFS is used as the storage layer to store S3 object data. Within HSFS, objects can be stored and protected in two ways; replicated and erasure coded.

SERVICE NAME	DESCRIPTION AND ROLE
Cassandra DB Services	The Cloudian HyperStore system uses the open source storage platform Cassandra to store several types of data. The Cloudian HyperStore system creates and uses several distinct “key spaces” within Cassandra. Note that S3 client applications do not access Cassandra databases directly. All S3 access is mediated through the S3 Service.
Redis DB Services	The Cloudian HyperStore system uses the lightweight, open source Redis key-value data store to store a variety of data that supports Cloudian HyperStore S3 service features. There are two types of Redis databases (DBs) in a Cloudian HyperStore deployment: The Redis Credentials DB stores user credentials and additional S3 operation supporting data such as multi-part upload session information and public URL access counters. The Redis QoS DB stores user-level and group-level QoS settings that have been established by system administrators. The DB is also used to keep count of user requests, so that QoS limits can be enforced by the system. The S3 Service, Administrative Service, and Cloudian HyperStore Service are the clients to these two Redis databases. Communication is through a protocol called Redis Serialization Protocol (RESP).
Administrative Service	The Cloudian HyperStore Administrative Service implements a RESTful HTTP API through which you can perform administrative operations such as: • Provisioning groups and users. • Managing QoS controls. • Creating and managing rating plans. • Generating usage data reports. The Cloudian Management Console (CMC) is a client to the Administrative Service. Building your own client is also an option. The Administrative Service is closely integrated with the S3 Service. Both leverage Jetty technology and are installed together. Both are started and stopped together by the same commands.

SERVICE NAME	DESCRIPTION AND ROLE
Cloudian Management Console (CMC)	The CMC is a web-based user interface for Cloudian HyperStore system administrators, group administrators, and end users. The functionality available through the CMC depends on the user type associated with a user's login ID (system administrative, group administrative, or regular user). As a Cloudian HyperStore system administrator, you can use the CMC to perform tasks such as: • Provisioning groups and users. • Managing QoS controls. • Creating and managing rating plans. • Generating usage data reports. • Generating bills. • Viewing and managing users' stored data objects. • Setting access control rights on users' buckets and stored objects. • Monitor storage system utilization and performance Group administrators can perform a more limited range of administrative tasks pertaining to their own group. Regular users can perform S3 operations such as creating and deleting bucket and uploading and downloading S3 objects. The CMC acts as a client to the Administrative Service and the S3 Service.
Supporting Services	Services that play a supporting role for the Cloudian HyperStore system include: Cloudian Monitoring Agent — The Cloudian Monitoring Agent runs on each Cloudian HyperStore node and monitors node health and performance statistics. The Agent also plays a role in the triggering of event notification emails to system administrators. System and node statistics are viewable through the CMC; and you can configure event notification rules through the CMC as well. Cloudian Monitoring Data Collector — The Cloudian Monitoring Data Collector runs on one node in each of your service regions, and regularly collects data from the Monitoring Agents. The Monitoring Collector writes its collected node health statistics to Cassandra's "Monitoring" key space. Puppet — As part of the Cloudian HyperStore software installation, the Cloudian HyperStore installer installs the open source version of Puppet and uses it to implement initial Cloudian HyperStore system configuration. Cloudian HyperStore also uses Puppet for support of ongoing configuration management. Dnsmasq — Dnsmasq is an optional lightweight domain resolution utility suitable for small networks. This utility is bundled with Cloudian HyperStore software distribution. The Cloudian HyperStore interactive installation wizard automatically installs and configures dnsmasq based on user input to resolve Cloudian HyperStore service domains; specifically, the S3 service domain, the S3 website endpoint domain, and the CMC domain.

CLOUDIAN HYPERSTORE OPERATIONS

In this section, we will review how Cloudian HyperStore works and some of the main functional areas:

- User Management
- Policy-based Data Protection
- Multi Datacenter Distribution
- Smart Repair
- Auto-tiering
- Per-bucket statistics
- Quality of Service (QoS)

USER MANAGEMENT

User groups and individual users can be provisioned through the Cloudian HyperStore Administrative API or through the CMC, these are the users or groups that are authorized to use the Cloudian HyperStore S3 service. The Cloudian HyperStore system can support millions of groups and users.

Groups are provisioned first, and then once one or more groups exist you can add individual users to each group. All users must belong to a group. As a HyperStore system administrator you can act on groups in a variety of ways:

- Each group can be assigned QoS limits that will enforce upper bounds on the service usage levels. Each group can also be assigned default user-level QoS controls that will limit the service usage of individual users within the group. If desired, assign and enforce per-user QoS limits that will supersede the default.
- You can generate service usage reports for groups (and also for individual users).
- Each group can be assigned a default rating plan which will determine how users in that group will be charged for Cloudian HyperStore service usage. If desired, you can also assign per-user rating plans that will supersede this default.
- Hyperstore offers selective IAM user supports (IAM API "actions"). For the list of supported actions, please refer to the online help.
- HyperStore system administrators cannot access and use the IAM API under their system administrator account. Only HyperStore group administrators and HyperStore regular users can use the IAM API functions to create IAM groups and IAM users and perform other IAM tasks.

POLICY-BASED DATA PROTECTION

Central to Cloudian’s data protection are its Storage Policies. These policies are ways of protecting data so that it’s durable and highly available to users. The Cloudian HyperStore system lets you pre-configure one or more storage policies. Users have storage policy choices for each bucket that they create. This means each bucket can have a different protection scheme. This allows for more efficient storage utilization and provides a level of protection based on the importance of the data.

Note that users cannot create buckets until you have created at least one storage policy. For each storage policy that you create you can choose from either of two data protection methods:

- **Replication** — with replication, a configurable number of copies of each data object are maintained in the system, and each copy is stored on a different node. For example, with 3X replication 3 copies of each object are stored, with each copy on a different node. This scheme is analogous to RAID 1 or RAID 10 for traditional SAN and NAS storage systems.
- **Erasure Coding** — with erasure coding, each object is encoded into a configurable number (known as the “k” value) of data fragments plus a configurable number (the “m” value) of redundant parity fragments. Each

fragment is stored on a different node, and the object can be decoded from any "k" number of fragments. For example, in a 4:2 erasure coding configuration (4 data fragments plus 2 parity fragments), each object is encoded into a total of 6 fragments which are stored on 6 different nodes, and the object can be decoded and read so long as any 4 of those 6 fragments are available. This scheme is analogous to RAID 5 or RAID 6 for traditional SAN and NAS storage systems. Erasure coding requires a minimum of 6 nodes.

In general, erasure coding requires less storage overhead (the amount of storage required for data redundancy) and results in somewhat longer request latency than replication. Erasure coding is best suited to large objects over a low latency network.

If your Cloudian HyperStore system spans multiple datacenters, you can also choose how data is allocated across your datacenters for each storage policy. For example, you could have a storage policy that for each S3 object, store 3 replicas of the object in each of your datacenters; and a second storage policy that erasure codes objects and stores them in just one particular datacenter.

SUPPORTED ERASURE CODING CONFIGURATIONS

Cloudian HyperStore supports many different EC configurations. These include Single Data Center (DC) EC, replicated EC and distributed EC configurations:

SINGLE DC EC:

This configuration requires a minimum 6 nodes across a single Data Center. In order to support the minimum data and parity fragments of (4+2) where 2 is the parity this is a requirement. Below is a table of the default EC configuration and the minimum number of nodes required in a single DC.
Note: Cloudian also supports 5 nodes EC as EC3+2. This requires a customized storage policy.

Nodes in the DC	EC
6	4+2
8	6+2
10	8+2
12	9+3
16	12+4

Table 1 – Single DC, EC Configuration

- 4+2 — each object will be encoded into 4 data fragments plus 2 parity fragments, with each fragment stored on a different node. Objects can be read so long as any 4 of the 6 nodes are available.
- 6+2 — each object will be encoded into 6 data fragments and 2 parity fragments, with each fragment stored on a different node. Objects can be read so long as any of the 6 of the 8 nodes are available.
- 9+3 — each object will be encoded into 9 data fragments plus 3 parity fragments, with each fragment stored on a different node. Objects can be read so long as any 9 of the 12 nodes are available.
- 12+4 — each object will be encoded into 12 data fragments plus 4 parity fragments, with each fragment stored on a different node. Objects can be read so long as any 9 of the 12 nodes are available.

Replicated EC:

This configuration requires a minimum of two DCs. Each DC must consist of minimum of 3 nodes each. This supports the minimum data and parity fragments of (2+1) where 1 is the parity. Below is a table of the default replicated EC configuration and the default number of nodes per DC.

Nodes Total	DC1	DC2	EC
6	3	3	2+1
12	6	6	4+2
16	8	8	6+2
20	10	10	8+2
24	12	12	9+3

Table 2 – Multiple DC, Replicated EC Configuration

Each object is encoded into equal parts and parity fragments are replicated on each node. Each DC is a mirror image and is analogous to a RAID10 mirror. For configurations greater than 2 DC using distributed EC configuration is recommended.

- 2+1 — each object will be encoded into 2 data fragments plus 1 parity fragment, with each fragment stored on a different node. Objects can be read so long as any 2 of the 3 fragments are available in at least 1 of the DCs.
- 4+2 — each object will be encoded into 4 data fragments plus 2 parity fragments, with each fragment stored on a different node. Objects can be read so long as any 4 of the 6 fragments are available in at least 1 of the DCs.
- 6+2 — each object will be encoded into 6 data fragments and 2 parity fragments, with each fragment stored on a different node. Objects can be read so long as any of the 6 of the 8 fragments are available in at least 1 of the DCs.
- 8+2 — each object will be encoded into 8 data fragments and 2 parity fragments, with each fragment stored on a different node. Objects can be read so long as any of the 8 of the 10 fragments are available in at least 1 of the DCs.
- 9+3 — each object will be encoded into 9 data fragments plus 3 parity fragments, with each fragment stored on a different node. Objects can be read so long as any 9 of the 12 fragments are available in at least 1 of the DCs.

The choice among these three supported EC configurations is largely a matter of how many Cloudian HyperStore nodes you have in the datacenter and the desired protection level. For a replicated EC configuration, a minimum of 3 nodes per DC are required.

4+2 EC provides a higher degree of protection and availability than 2+1 EC (since with 4+2 EC, objects can be read/written even if 2 of the involved nodes or disks are inaccessible) while delivering the same level of storage efficiency (both 2+1 and 4+2 require 50% storage overhead — the parity fragments as a percentage of data fragments). So, 4+2 is preferable to 2+1 if you have at least 6 Cloudian HyperStore nodes in the datacenter.

Likewise, compared to a 4+2 configuration, 9+3 EC provides additional resiliency against nodes or disks being unavailable, while delivering a higher level of storage efficiency (3/9 = only 33% overhead). So, if you have at least 12 Cloudian HyperStore nodes, 9+3 EC is an attractive option. The cost, compared to a smaller number of fragments, is a modest increase in the time that it takes to decode objects.

Distributed EC:

Cloudian's Distributed EC solution implements the new ISA-L Erasure Codes which is vectorized and fast. ISA-L is the Intel library containing functions to improve erasure coding.

The Cloudian Distributed Data Center with EC configuration requires a minimum of 3 Data Centers with 4 nodes each for a total of 12 nodes. Distributed EC configuration offers the same level of protection as the replicated EC configuration with 50% less storage required. If the number of DCs are 3 or more, this configuration is recommended.

For 12 nodes, each object will be encoded into 12 data fragments plus 5 parity fragments, with each fragment stored on a different node. Objects can be read so long as any 7 of the 12 fragments are available across all of the DCs. This configuration can withstand the loss up to 5 nodes. Note that the performance is sensitive to the network latency.

Below is a table of defaults for Distributed Data Centers with EC configurations:

Nodes Total	Number of DC	Number of nodes per DC	EC	Fragments
12	3	4	7+5	7 per DC
48	4	12	8+4	8 per DC
50	5	10	6+4	6 per DC
72	6	12	7+5,8+4	7, 8 per DC
98	7	14	10+4	10 per DC
128	8	16	10+6	8 per DC
162	9	18	10+8	9 per DC

Table 3 – Multiple DC, Distributed EC Configuration

Cloudian also supports customizable EC to meet specific needs by setting a flag. Please contact your local sales team if you are interested in an EC configuration which is not listed in the defaults.

SMART REPAIR

Smart repair ensures data integrity and availability through proactive actions. Replicated data in a Cloudian HyperStore storage cluster is assessed and updated regularly to ensure that each data object has the configured number of replicas across the cluster, and that all replicas are current. If EC is enabled in a storage policy, it's also important that erasure coded data be regularly evaluated and repaired so that each object has the correct number of fragments throughout the cluster and all fragments are up-to-date. These actions are key foundation of data durability. To achieve true data durability, smart repair occurs at different times during an object lifetime. Cloudian HyperStore uses three smart repair mechanisms; repair-on-read, proactive repair and auto-repair.

Cloudian HyperStore uses a repair-on-read feature. When a read request is processed for a replicated object in the Cloudian HyperStore File System, all replicas of the object are checked, and any missing or out-of-date replicas are replaced or updated. A similar process is performed for EC objects and for metadata in Cassandra. This check is applied randomly to about 20% of EC object reads and about 10% of Cassandra data reads for each request. Since this mechanism repairs only data that is read. It is necessary to have additional mechanisms that regularly checks all data in the system and implement repairs as needed.

HyperStore proactive repair works by reading a list of failed write attempts for objects from Cassandra when a node was unavailable. Working from this object list, proactive repair streams in any locally missing replicas by copying them from nodes where they do exist. For erasure coded objects, the proactive repair process re-generates the locally missing fragment.

Third, regular repair of all data is accomplished by the Cloudian HyperStore auto-repair feature which automatically executes repair jobs on each node in the cluster on scheduled basis. The schedule is customizable by an admin if desired. Separate repair jobs are run for Cassandra data. That is, object metadata and service usage data that's replicated across the cluster and stored in Cassandra, replica data which is S3 object data that’s replicated across the cluster and stored in the Cloudian HyperStore File System (HSFS) and finally EC data which is erasure coded S3 object fragments that are distributed across the cluster and stored in the HSFS.

AUTO TIERING

As mentioned earlier, the Cloudian HyperStore system supports an auto-tiering feature whereby objects can be automatically moved from a source Cloudian HyperStore storage to a private or public destination storage system on a predefined schedule. Cloudian HyperStore also has a bridge feature where objects can be tiered immediately. Auto-tiering is configurable on a per-bucket basis and can be configured to use popular public cloud sites as well as on-premise systems as a destination. Auto-tiering is configurable on a per-bucket basis. Cloudian HyperStore administrators can specify:

- The auto-tiering destination such as a public cloud storage site or on-premise storage system
- Object filters. For example, auto-tiering can apply to all objects in the bucket or only to objects for which the full object name starts with a particular prefix
- The tiering schedule, which can be implemented in any of these ways. Move objects X number of days after they’re created. Move objects if they go X number of days without being accessed. Move objects on a fixed date — such as December 31, 2018
- Bridge mode which defines moving objects immediately after they're created

The destination storage system can be any of the following locations:

- Amazon S3
- Amazon Glacier
- Google Storage Cloud
- HyperStore region or system
- Different S3-compliant system
- In addition, HyperStore supports auto-tiering to Microsoft Azure and Spectra Logic Black Pearl.

By default, auto-tiering is not enabled for any bucket, and the CMC functionality for activating auto-tiering is obscured. Also note auto-tiering restrictions apply to some of these destinations. For more information about configuring auto-tiering in the system and on individual buckets please see the online user guide.

PER-BUCKET USAGE TRACKING

Cloudian Hyperstore enables storage capacity and performance monitoring on a per-bucket basis. To enable per-bucket usage statistics of all types - Storage Bytes, Storage Objects, HTTP Requests, Bytes IN, Bytes OUT, use the configuration template. Once turned on, push the change out to the cluster and restart the S3 Service. For detailed instructions please see the online user guide.

Note that once you enable this feature, this type of usage data will be tracked and available for reporting from that point in time forward. There will not be any per-bucket usage data from prior to that time. Be aware that enabling this feature results in additional data being stored in Cassandra, and additional work for the cron jobs that roll up usage data into hourly, daily, and monthly aggregates.

QUALITY OF SERVICE (QOS)

The Cloudian HyperStore system supports flexible user-level and group-level Quality of Service (QoS) settings as follows:

- User QoS settings place upper limits on service usage by individual users.
- Group QoS settings place upper limits on aggregate service usage by entire user groups.

To enforces QoS settings, the HyperStore system rejects S3 requests that would result in a user (or a user's group) exceeding the allowed service usage level. Several types of service usage metrics can be configured for QoS controls:

- Storage quota, by number of KBs.
- Storage quota, by number of objects.
- Peak HTTP request rate, in requests per minute. The user is not allowed more than this many requests in a 60 second interval.
- Peak data upload rate, in KBs per minute.
- Peak data download rate, in KBs per minute.

RICH OBJECT METADATA

Like Amazon S3, Cloudian HyperStore allows for rich metadata to be associated with each stored object. This unlocks intelligence in the data, enabling deep search applications, data science techniques, and machine learning. The metadata is also replicated across the nodes for redundancy. The system allows for user-defined object metadata as well as system-defined object metadata.

S3 object metadata is subject to the same configurable replication factor as S3 object data. This replication factor determines how many replicas of each S3 object will be maintained in each of the configured data centers. The system will also maintain this same number of replicas of the metadata for each object. For example, if you configure a storage policy so that each S3 object is replicated in the HyperStore File System on three nodes in DC1 and two nodes in DC2, then each object's metadata will also be replicated in Cassandra on three nodes in DC1 and two nodes in DC2.

The Cloudian HyperStore system supports the Amazon S3 API methods that enable client applications to set user-defined object metadata as an object is being stored, as well as the S3 API methods that facilitate the subsequent retrieval of a specified object's metadata with or without the object itself. Additionally, the HyperStore extends the Amazon S3 API by allowing client applications to retrieve the user -defined metadata associated with all of the objects in a specified bucket. Cloudian HyperStore provides system operators the option to configure different consistency requirements for replicated object metadata than for the replicated S3 object data itself within a storage policy.

CLOUDIAN HYPERSTORE INTERNALS

OBJECT METADATA

In Cloudian HyperStore, object metadata is stored in Cassandra, specifically, in the CLOUDIAN_METADATA and CLOUDIAN_OBJMETADATA column families within each of the "UserData_<policyid>" key spaces. For each stored S3 object, whether it's a replicated object or an erasure-coded object, the object's metadata is written to both the CLOUDIAN_METADATA column family and the CLOUDIAN_OBJMETADATA column family. The CLOUDIAN_METADATA column family is organized as one row per S3 storage bucket, while the CLOUDIAN_ OBJMETADATA column family is organized as one row per object. The Cloudian HyperStore system uses the two different column families for different purposes. For example, the CLOUDIAN_OBJMETADATA column family is optimized for reads of a given object's metadata.

For each object, Cloudian HyperStore maintains a variety of system-defined object metadata including (but not limited to) the following:

- Creation time
- Last modified time
- Last accessed time
- Size
- ACL information
- Version, if applicable
- Public URL, if applicable
- Compression type, if applicable
- Encryption key, if applicable
- Auto-tiering state, if applicable

By default, objects do not have user-defined metadata associated with them, but the Cloudian HyperStore storage schema supports storing user-defined object metadata together with system-defined object metadata, in those same two column families.

CLOUDIAN HYPERSTORE VNODES

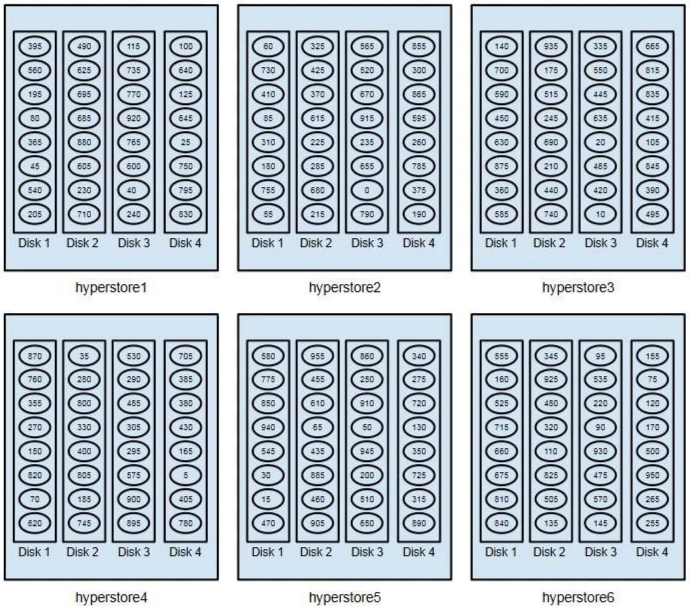
S3 object placement and replication within a Cloudian HyperStore geo-cluster is based on a consistent hashing scheme that utilizes an integer token space ranging from 0 to 2127 -1. Integer tokens from within this token space are assigned to the Cloudian HyperStore nodes. Then, a hash value is calculated for each S3 object as it is being uploaded to storage. The object is stored to the node that has been assigned the lowest token value higher than or equal to the object's hash value. Replication is implemented by also storing the object to the nodes that have been assigned the next-higher tokens.

VNODES IN DETAIL

Traditionally, hash-based storage schemes are assigned just one token per physical node. This optimized design assigns a large number of tokens (up to a maximum of 256) to each physical host. In essence, the storage cluster is composed of very many virtualized nodes. Multiple virtual nodes reside on each physical host.

The HyperStore system goes much further by assigning a different set of tokens (virtual nodes) to each disk on each physical host. With this architecture, each disk on a host is responsible for a different set of object replicas, and if a disk fails it affects only the object replicas on that one disk. Other disks in the host can continue operating and supporting their own data storage operations.

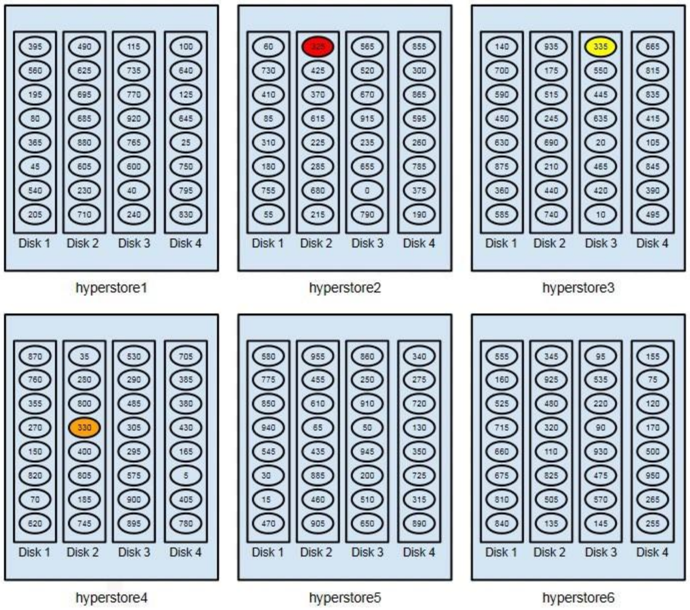
For example, consider a geo-cluster of six Cloudian HyperStore hosts each of which has four disks designated for S3 object storage. Suppose that each physical host is assigned 32 tokens. And suppose that there is a simplified token space ranging from 0 to 960, and the values of the 192 tokens in this system (six hosts times 32 tokens each) are 0, 5, 10, 15, 20, and so on up through 955.



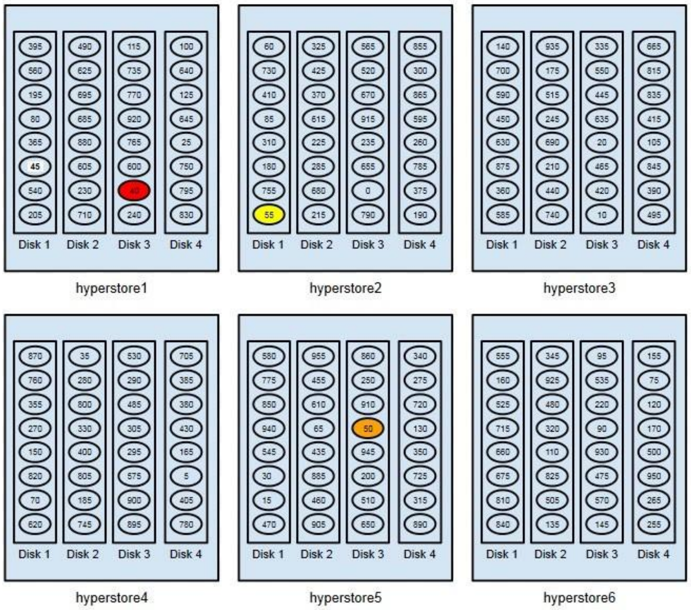
The illustration to the right shows one possible allocation of tokens across the cluster. Each host's 32 tokens are divided evenly across the four disks (eight tokens per disk), and that token assignment is randomized across the cluster.

Now further suppose that you've configured your Cloudian HyperStore system for 3X replication of S3 objects. And say that an S3 object is uploaded to the system and the hashing algorithm applied to the object name gives us a hash value of 322. The diagram below shows how three instances or replicas of the object will be stored across the cluster:

- With its object name hash value of 322, the primary replica of the object is stored where the 325 token is located. This is the lowest token value that's higher than or equal to the object hash value. The 325 token (highlighted in red in the diagram) is assigned to Cloudian HyperStore2:Disk2, so that's where the primary replica of the object is stored. Note that the primary replica has no functional primacy compared to other replicas; it's called that only because its placement is based simply on identifying the disk that's assigned the token range into which the object hash falls.
- The secondary replica is stored to the disk that's assigned the next-higher token (330, highlighted in orange), which is Cloudian HyperStore4:Disk2.
- The tertiary replica is stored to the disk that's assigned the next-higher token after that (335, in yellow), which is Cloudian HyperStore3:Disk3.



Working with the same cluster and simplified token space, we can next consider a second object replication example that illustrates an important vNode feature: no more than one of an object's replicas will be stored on the same physical host. Suppose an S3 object is uploaded to the system and the object name hash is 38. The next diagram shows how the object's three replicas are placed.



- The primary replica is stored to the disk where token 40 is — Cloudian HyperStore1:Disk3 (red highlight).
- The next-higher token — 45 (with high-contrast label) — is on a different disk (Disk1) on the same physical host as token 40, where the Cloudian HyperStore system is placing the primary replica. Because it's on the same physical host, the system skips over token 45 and places the object's secondary replica where token 50 is — Cloudian HyperStore5:Disk3 (orange highlight).
- The tertiary replica is placed on Cloudian HyperStore2:Disk1, where token 55 is (yellow highlight).

SERVER-SIDE ENCRYPTION

Like Amazon S3, the Cloudian HyperStore system supports server-side encryption (SSE) to protect the confidentiality of data at rest. The Cloudian HyperStore system can perform the encryption, and subsequent decryption upon object retrieval. Like Amazon S3, this is either with a system-generated encryption key (regular SSE) or a customer-provided encryption key (SSE-C).

- The object upload and download requests must be submitted to the system via HTTPS, not regular HTTP.
- The system does not store a copy of the encryption key.
- The user is responsible for managing the SSE-C encryption key. If an object is uploaded to Cloudian HyperStore system and encrypted with a user-provided key, the user will need to provide that same key when later requesting to download the object. If the user loses the key, the encrypted object will not be downloadable.

WORM

The HyperStore mechanism for implementing WORM is called a bucket lock. To apply a lock to a bucket, the bucket must first have versioning enabled. When versioning is enabled on a bucket, each version of objects in the bucket is retained. For example, if you upload a newly created document, and then on two subsequent occasions you revise the document and upload it again, the system will retain all three versions of the document.

To apply a bucket lock to a versioning-enabled bucket is two-step process. First you initiate the lock. This includes defining a retention period to apply to objects in the bucket. Once the lock is initiated, there are 24 hours to complete the lock. As soon as the bucket lock initiates, objects in the bucket become protected. Note, if older than defined in the retention period the objects are subject to deletion. During this initial 24-hour period, there is an option to remove the lock from the bucket. If no action is taken, the system will automatically remove the lock from the bucket after the 24 hours expire. The next step is to complete the lock. This must be done during the initial 24-hour period. Once completed, then the lock can never be removed from the bucket or modified. For more information see the online user guide.

BUCKET LOCKS IN DETAIL

Once a bucket lock is permanent, then each object in the bucket is protected against deletion until the object age exceeds the defined retention period. This is enforced at the S3 API level. If an S3 delete request for an object in the bucket is received, the system checks the object creation date-time stamp, the date-time of object upload, and adds to that the retention period defined in the bucket lock policy. If the object is still within the retention period, then the S3 delete request is rejected. So, if an object's creation date-timestamp is from December 8, 2017 and the bucket's lock policy mandates a retention period of five years, then the system will not allow the object to be deleted through the S3 API until after December 8, 2022.

It's important to note that the retention period is applied on a per-object basis, commencing from object creation date-time. For example, if you were to execute a bucket lock on January 15, 2018, with a 5-year retention period, all objects in the bucket would not necessarily be eligible for deletion after January 15, 2023. Consider the following. An existing object that had been uploaded to the bucket on April 4, 2017 would be eligible for deletion after April 4, 2022. An object that gets uploaded to the bucket on September 9, 2018, after the lock is in place, would be eligible for deletion after September 9, 2023.

In a versioning-enabled bucket note that the retention period is applied separately to each object version based on the creation date-time stamp. For example, if there are two version of an object, version1 and version2, with version1 being the older version, version1 of the object would become eligible for deletion before version2.

ACCESS PROTOCOLS AND APPLICATIONS

Cloudian supports multiple access protocols to your data. A RESTful interface (based on S3), NFS and SMB3. Depending on your application you can choose the appropriate access protocol.

100% S3 NATIVE INTERFACE

Cloudian HyperStore is the only storage platform that offers 100% S3 native compatibility, as well as the only storage platform in the advanced compatibility tier that allows developers continued use of Amazon’s S3 Software Development Kit (SDK). By supporting native S3 API calls, developers can significantly ease their workloads by not changing SDK’s or API’s. Additionally, Cloudian is the only storage platform that automatically tiers data between on-premises cloud deployments and Amazon’s S3 public cloud while representing the cloud ecosystem under a single name space. With these features, Cloudian HyperStore is the most compatible storage platform for S3 on-premises and hybrid cloud deployments.

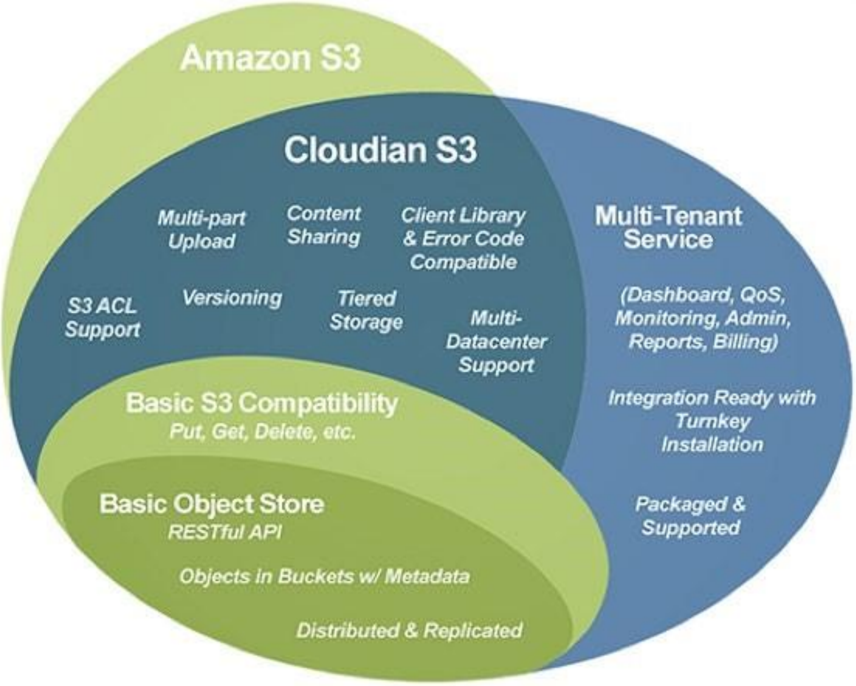


Figure 3 – Amazon S3 and Cloudian S3 Compatibility

Amazon S3 commands twice the market share of all its closest competitors combined and it will likely be the storage platform of choice for on-premises hybrid or private cloud deployments. Companies and developers implementing S3 depend significantly on compatibility with S3. With no standards enforced for claiming S3 compatibility, choosing the right storage platform can be tenuous.

When looking at deploying an open-hybrid cloud and/or moving data between storage and the private cloud, it is of utmost importance to understand the level of S3 compatibility a storage platform claims versus its actual compatibility. S3 is quickly becoming the object storage standard and will not disappear anytime soon. Choosing the right storage platform for the hybrid or private cloud can save organizations money and shave months off of the time to deploy. In essence, compatibility matters.

To make S3 simple for applications to access, Amazon continues to refine the operations available to applications through the S3 API. The API enables operations to be performed on the S3 Service, S3 Buckets, and S3 Objects; there are more than 50 operations available today. Compatibility is based on a storage platform’s ability to perform some, many, or all of the S3 API features.

ADVANCED S3 COMPATIBILITY

For organizations and developers that want assurance that their applications are S3 compatible and/ or all + S3 compatible applications will continue to work seamlessly with their hybrid or on-premises cloud, choosing a storage platform that boasts advanced compatibility with the S3 API is vital. Of the 50+ operations available through the S3 API, more than 25 of them are considered advanced. To be considered compatible with S3, a storage platform needs work with the majority of the advanced operations. Cloudian Hyperstore is continually tested against all S3 operations to maintain 100% compatibility.

EXTENSIONS TO THE S3 API

Not only does Cloudian HyperStore offer 100% S3 advanced compatibility, it also extends the S3 API to gain additional functionality. This section will explain each of the extensions added to Cloudian HyperStore. For more information on implementing these features, please refer to the Cloudian HyperStore Configuration guide and the Cloudian HyperStore S3 Development guides.

LARGE S3 OBJECTS

This extension of the S3 API enables S3 clients to configure on a per-bucket basis how to store and protect "large" S3 objects. The options include storing and replicating such objects in the Cloudian HSFS or storing and protecting such objects through erasure coding. This S3 API extension for bucket configuration also enables clients to specify the size threshold above which an object is to be processed as a "large" object.

VIRTUAL BUCKETS

A virtual bucket is a bucket of unlimited size for which object storage spans multiple regions. The virtual bucket feature is applicable only for a multi-region Cloudian HyperStore deployment. Its purpose is to accommodate buckets with larger capacities than service providers may wish to allow in a single region.

After an S3 client creates a bucket in a particular region, the client can request that the bucket be implemented as a virtual bucket. An S3 extension API method is available to enable (or subsequently disable) virtualization on a specific bucket. When a bucket is virtualized, objects PUT into that bucket may be stored in the same region in which the bucket was created, or in any other region in the Cloudian HyperStore service deployment.

Subsequent requests to retrieve (or delete) an object stored in a virtual bucket will be routed to the correct region by the S3 server receiving the request. After request processing, the request-receiving S3 Server will return a response to the client. For purposes of QoS enforcement and billing, all usage data for the bucket is tracked at the region in which the bucket was created. This is regardless of which region a virtualized bucket's objects are stored.

PER BUCKET WORM POLICIES

HyperStore supports applying a "Write Once, Read Many" (WORM) policy to a bucket, so that data in a bucket is kept in unaltered form for a defined retention period.

CANNED ACL GROUP-READ

Canned ACL Group-read allows read access to everyone in the object owner's group. To grant access to groups other than the requester's group, while using standard Amazon S3 methods for assigning privileges to grantee, specify "<groupID>" as the grantee. When separate PUT ACL requests grant permissions both to a group and to an individual user within that group, the user gets the broader of the two permission grants. For example, if the group is granted full control, and a user within the group is granted read, the user gets full control.

RETURN USER-DEFINED OBJECT METADATA

This extension is enabled by the optional extension URI parameter: meta=true. This extension enables developers to return user-defined object metadata with the GET Bucket response. Without this Cloudian HyperStore extension, the GET Bucket method returns only system metadata, not user-defined metadata. See Figure 4, for a sample output of an HTTP request viewing the object metadata.

TRANSITION RULES

You can configure schedule-based automatic transition (also known as "auto-tiering") from Cloudian HyperStore storage to Amazon S3 storage, Amazon Glacier storage, Storage in a different Cloudian HyperStore service region.

USER MANAGEMENT

The user management implementation allows administrators to retrieve a user's profile, list users, create new users, manage S3 credentials, and configure a rating plan.

GROUP MANAGEMENT

The group management implementation allows administrators to manage a group by retrieving information and provides basic creation and deletion functions. Like user management, the API also enables rating plan management on a group level.

PUBLIC URL SERVICE

This set of Administrative API methods enable a “public URL” feature whereby URLs can be assigned to stored objects that enable the public to access those objects through a web browser, without having to use the S3 protocol.

QOS LIMITS SERVICE

These API methods allow administrators to retrieve the Quality of Service (QoS) settings for users or groups.

USAGE REPORTING

Allows administrators to retrieve S3 service usage data for a Cloudian HyperStore user or for a user group. Cloudian HyperStore usage reporting complies with Amazon S3 in that data transfer and storage activity is always attributed to the bucket owner, regardless of who owns individual objects within the bucket or who submits object-related requests.

RATING PLAN SERVICE

The Cloudian HyperStore Rating Plan Service allows administrators to retrieve information about a particular rating plan. Also, there are methods available to update and delete rating plans.

BILLING SERVICE

API implementation allows access to retrieve a user’s bill after it is generated. Bills are available for retrieval only for a completed month.

BILLING WHITELIST SERVICE

Administrators can specify a list of IP addresses or subnets that are allowed to have free S3 traffic with the Cloudian HyperStore storage service. For S3 requests originating from addresses on this “whitelist”, a special rating plan is used that applies zero charge to all the traffic-related pricing metrics.

SYSTEM SERVICES

The system services API methods allow administrators to gather license information and system attributes. Also, audit data can be retrieved to gather usage information on the cluster.

DEVELOPING S3 APPLICATIONS

In nearly every way, developing a client application for the Cloudian HyperStore storage service is the same as developing a client application for Amazon S3. Consequently, when designing and building S3 applications for the Cloudian HyperStore service you can leverage the wealth of resources available to Amazon S3 developers.

The best place to turn for resources for developing Amazon S3 and Cloudian HyperStore S3 applications is the Amazon S3 web site. Through that site, Amazon Web Services (AWS) Developer Centers are available for a variety of development technologies:

- AWS Java Developer Center
- AWS Windows/.NET Developer Center
- AWS PHP Developer Center
- AWS Python Developer Center
- AWS Ruby Developer Center

- AWS Developer Centers include SDKs, community libraries, “Getting Started” guides, and tips and tricks.

Another good Amazon resource is the archive of Articles & Tutorials. The archive includes general articles such as “Best Practices for Using Amazon S3” as well as articles and tutorials relating to specific development technologies. Yet another helpful Amazon resource is the archive of Sample Code & Libraries, which can be found here: http://aws.amazon.com/articles?_encoding=UTF8&jiveRedirect=1

CLOUDIAN HYPERSTORE MANAGEMENT

ONE SIMPLE WEB BASED GUI

The Cloudian Management Console (CMC) is a web-based user interface for Cloudian HyperStore system administrators, group administrators, and end users. The functionality available through the CMC depends on the user type associated with a user’s group membership and login ID. Roles include system administration group administration and ordinary users.

As a Cloudian HyperStore system administrator, the CMC can perform tasks such as:

- Provisioning groups and users.
- Managing quality of service (QoS) controls.
- Creating and managing rating plans.
- Generating usage data reports.
- Generating bills.
- Viewing and managing users’ stored data objects.
- Setting access control rights on users’ buckets and stored objects.

Group administrators can perform a more limited range of administrative tasks pertaining to their own group. Regular users can only perform S3 operations such as uploading and downloading S3 objects. The CMC is a client to the Administrative Service and the S3 Service.

CONFIGURABLE

The Cloudian HyperStore system has lots of options to meet your specific needs. Basic system configuration is implemented by the interactive Cloudian HyperStore installation script. For the majority of ongoing management, settings can be modified CMC. Beyond that, a wider range of settings can be modified by editing configuration file templates on the Puppet master node. Puppet is then used to propagate the changes throughout the Cloudian HyperStore cluster and then restarting the services from a simple menu selection.

AUTOMATED PROVSIONING OF USERS AND GROUPS

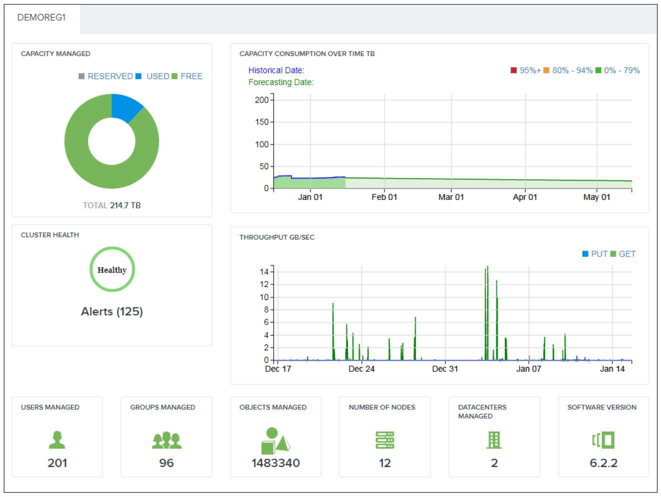
Cloudian HyperStore provides a RESTful HTTP API through which you can provision users and groups, manage rating plans and quality of service controls, and automate other administrative tasks. This Administrative API is supported by the Cloudian HyperStore Administrative Service which run all nodes with the HyperStore S3 Service. The HTTP listening port for the Administrative API is 18081.

Cloudian HyperStore Administrative API response payloads are JSON encoded. For POST or PUT requests that require a request payload, note the request payloads must be JSON encoded too.

VIEW SUMMARY FOR DETAILS

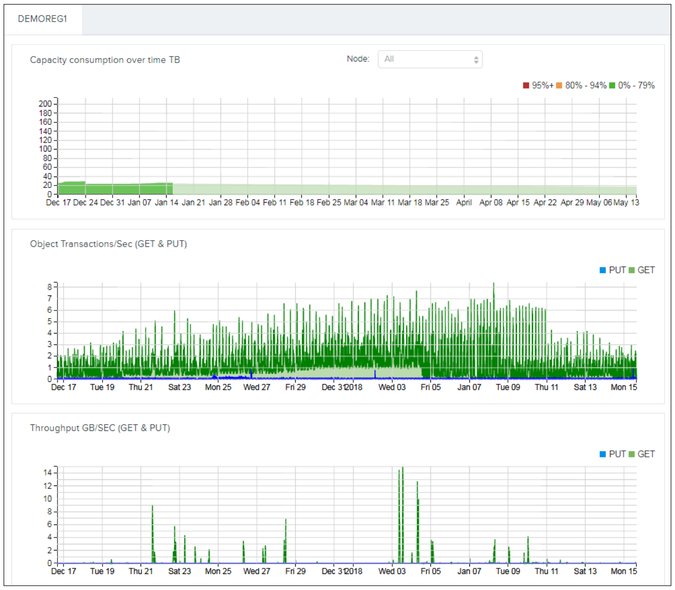
SIMPLE DASHBOARD

The CMC dashboard provides a high-level view of the status of your Cloudian HyperStore object storage service. If you have multiple service regions, there is a separate dashboard view for each region.



SINGLE CLUSTER USAGE & PERFORMANCE VIEW

View cluster usage graphs that cover the past 30 days of activity.


DEMOREG1
Capacity consumption over time TB
Node: All
■ 95%+ ■ 80% - 94% ■ 0% - 79%
200
180
160
140
120
100
80
60
40
20
0
Dec 17 Dec 24 Dec 31 Jan 07 Jan 14 Jan 21 Jan 28 Feb 04 Feb 11 Feb 18 Feb 25 Mar 04 Mar 11 Mar 18 Mar 25 April Apr 08 Apr 15 Apr 22 Apr 29 May 06 May 13
Object Transactions/Sec (GET & PUT)
■ PUT ■ GET
8
7
6
5
4
3
2
1
0
Dec 17 Tue 19 Thu 21 Sat 23 Mon 25 Wed 27 Fri 29 Dec 31 2018 Wed 03 Fri 05 Jan 07 Tue 09 Thu 11 Sat 13 Mon 15
Throughput GB/SEC (GET & PUT)
■ PUT ■ GET
14
12
10
8
6
4
2
0
Dec 17 Tue 19 Thu 21 Sat 23 Mon 25 Wed 27 Fri 29 Dec 31 2018 Wed 03 Fri 05 Jan 07 Tue 09 Thu 11 Sat 13 Mon 15

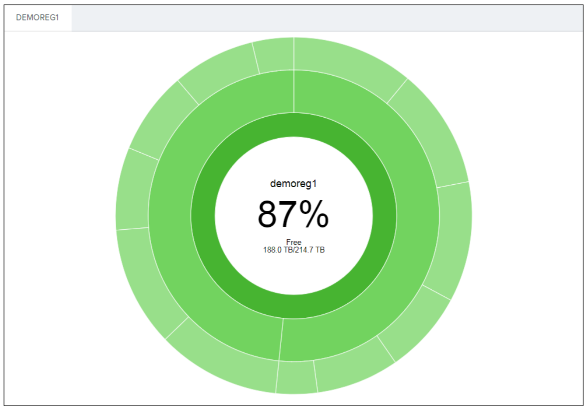
CAPACITY EXPLORER

With the CMC's **Capacity Explorer** page, you can view your available S3 object data storage capacity by region, by data center, and by node. First (if you have a multi-region system) choose a region tab at the top of the page. Then, in the graphical display:

- The **inner circle** represents the service region as a whole
- The **middle circle** has one segment for each data center in the region
- The **outer circle** has one segment for each node in each data center

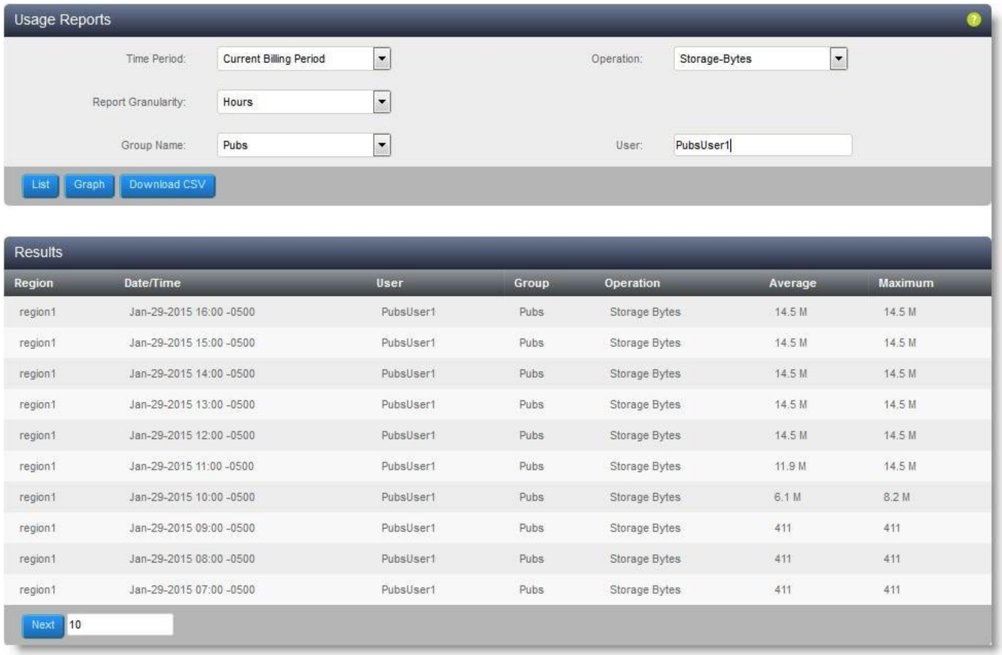
The circle segments are color-coded as follows:

- Green indicates that free space is 30% or more of total space for that region, data center, or node. (Slightly different shades of green are used merely to differentiate the concentric circles from each other. Green has the same meaning regardless of the particular shade of green.)
- Orange indicates that free space is between 10% and 29% of total space for that region, data center, or node.
- Red indicates that free space is less than 10% of total space for that region, data center, or node.



VIEW USER AND TENANT USAGE

In the CMC's "Usage by Users & Group" page you can generate service usage reports for individual users, for user groups, and for the system as a whole.



Usage Reports

Time Period: Current Billing Period

Operation: Storage-Bytes

Report Granularity: Hours

Group Name: Pubs

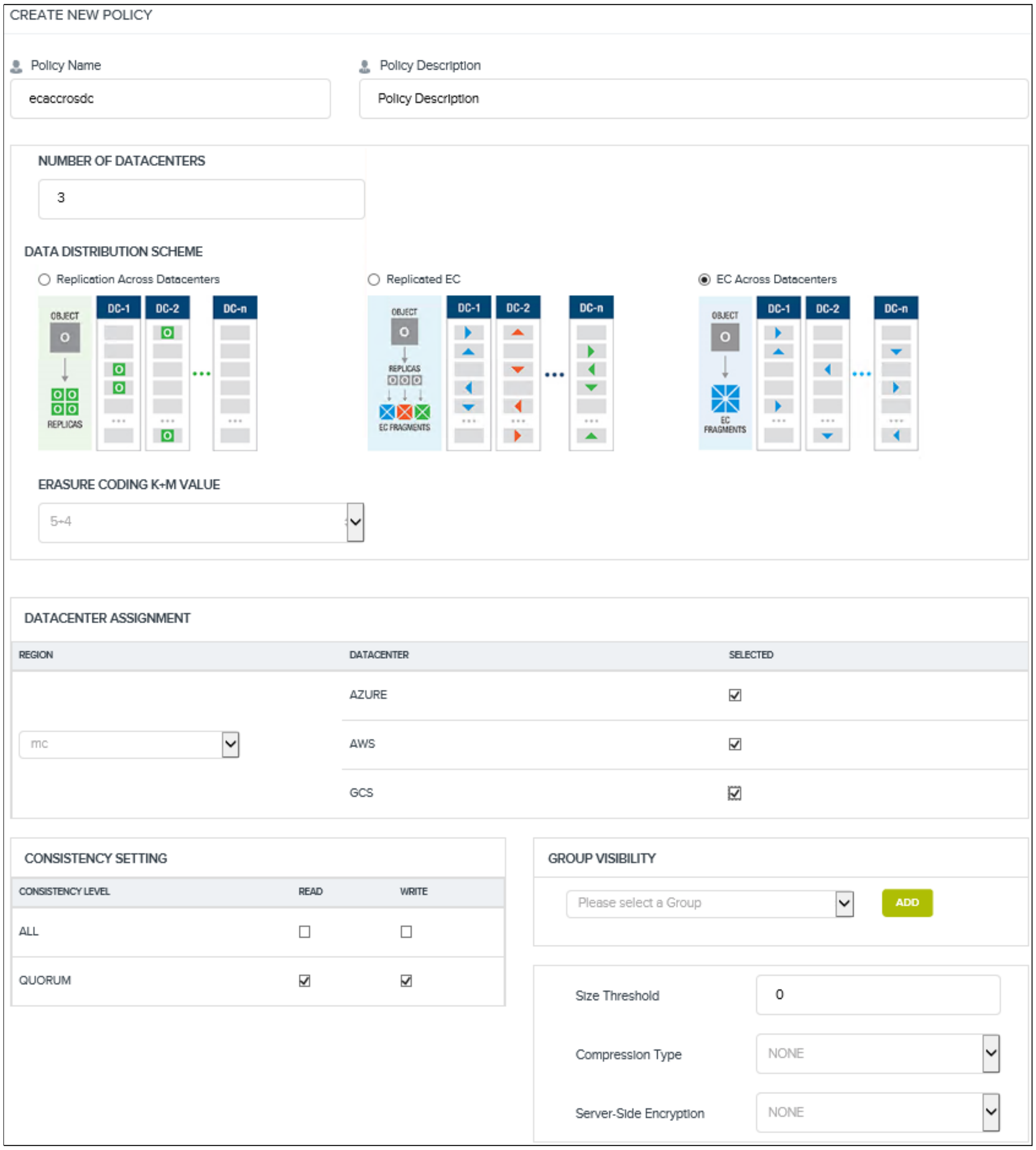
User: PubsUser1

Results

Region	Date/Time	User	Group	Operation	Average	Maximum
region1	Jan-29-2015 16:00 -0500	PubsUser1	Pubs	Storage Bytes	14.5 M	14.5 M
region1	Jan-29-2015 15:00 -0500	PubsUser1	Pubs	Storage Bytes	14.5 M	14.5 M
region1	Jan-29-2015 14:00 -0500	PubsUser1	Pubs	Storage Bytes	14.5 M	14.5 M
region1	Jan-29-2015 13:00 -0500	PubsUser1	Pubs	Storage Bytes	14.5 M	14.5 M
region1	Jan-29-2015 12:00 -0500	PubsUser1	Pubs	Storage Bytes	14.5 M	14.5 M
region1	Jan-29-2015 11:00 -0500	PubsUser1	Pubs	Storage Bytes	11.9 M	14.5 M
region1	Jan-29-2015 10:00 -0500	PubsUser1	Pubs	Storage Bytes	6.1 M	8.2 M
region1	Jan-29-2015 09:00 -0500	PubsUser1	Pubs	Storage Bytes	411	411
region1	Jan-29-2015 08:00 -0500	PubsUser1	Pubs	Storage Bytes	411	411
region1	Jan-29-2015 07:00 -0500	PubsUser1	Pubs	Storage Bytes	411	411

STORAGE POLICIES

Central to Cloudian's data protection are its storage policies. These polices protect data that ensure data durability and high availability to users. The Cloudian HyperStore system lets you pre-configure one or more storage policies and are applied on a per-bucket basis. Users when they create a new storage bucket can then choose which pre-configured storage policy to use to protect the data. For each storage policy that you create, choose from either of two data protection methods: replication or erasure coding.



CREATE NEW POLICY

Policy Name: ecaccrosdc

Policy Description: Policy Description

NUMBER OF DATACENTERS: 3

DATA DISTRIBUTION SCHEME

○ Replication Across Datacenters ○ Replicated EC ● EC Across Datacenters

ERASURE CODING K+M VALUE: 5+4

DATACENTER ASSIGNMENT

REGION	DATACENTER	SELECTED
mc	AZURE	☑
	AWS	☑
	GCS	☑

CONSISTENCY SETTING

CONSISTENCY LEVEL	READ	WRITE
ALL	□	□
QUORUM	☑	☑

GROUP VISIBILITY

Please select a Group ADD

Size Threshold: 0

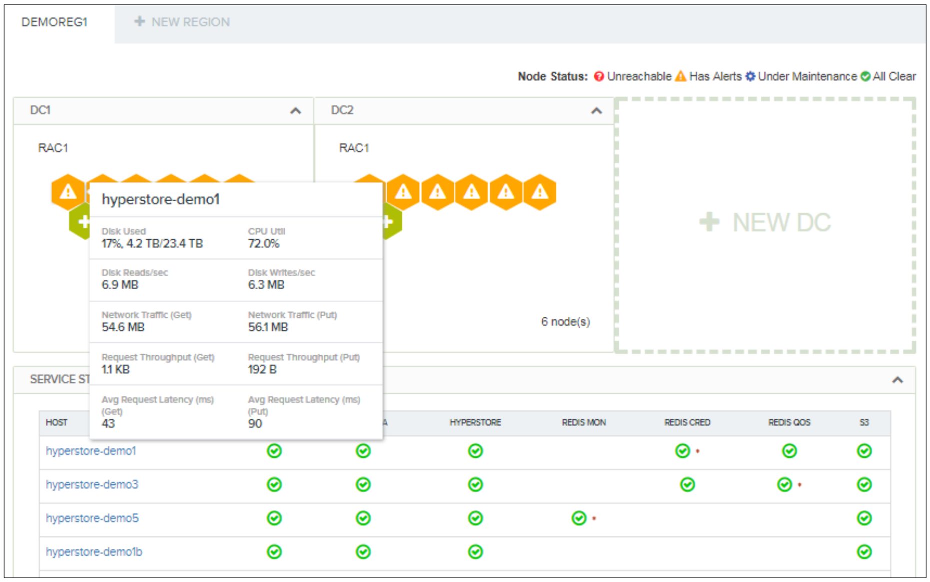
Compression Type: NONE

Server-Side Encryption: NONE

CLUSTER CONFIGURATION & MONTORING

MULTI-DATACENTER & REGION VIEW

The datacenters page displays a panel for each datacenter in your Cloudian HyperStore system. Each region appears in it's own tab. For example, the region below is DEMOREG1. For each datacenter, each Cloudian HyperStore node in the datacenter is represented by a cube under each region's tab.



Node Status: Unreachable Has Alerts Under Maintenance All Clear

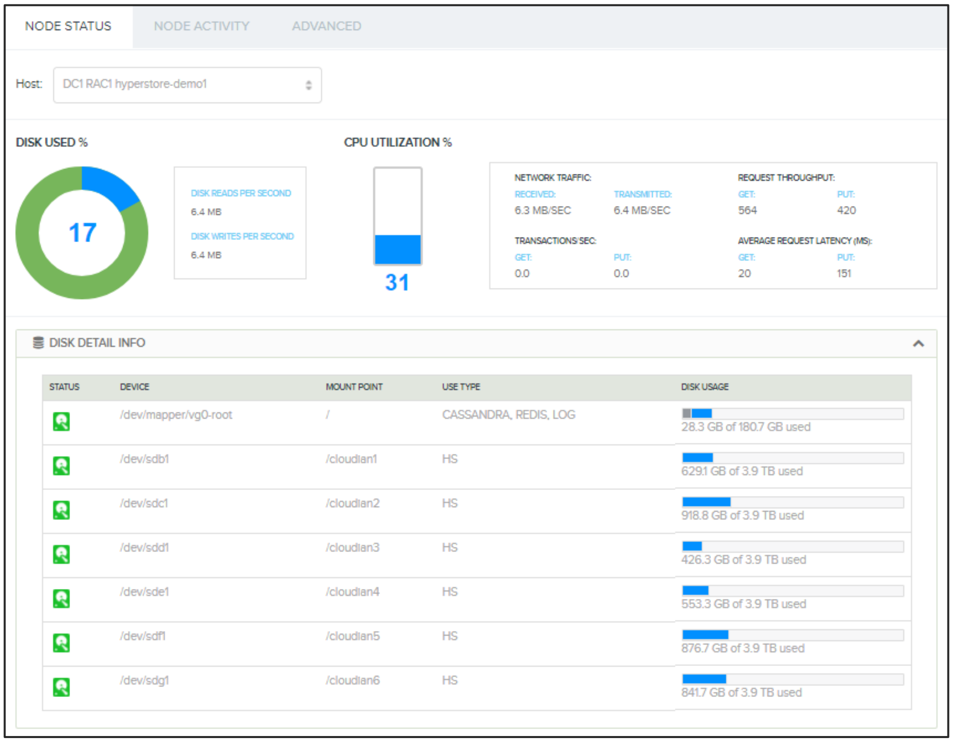
hyperstore-demo1

Disk Used 17%, 4.2 TB/23.4 TB	CPU Util 72.0%
Disk Reads/sec 6.9 MB	Disk Writes/sec 6.3 MB
Network Traffic (Get) 54.6 MB	Network Traffic (Put) 56.1 MB
Request Throughput (Get) 1.1 KB	Request Throughput (Put) 192 B
Avg Request Latency (ms) (Get) 43	Avg Request Latency (ms) (Put) 90

HOST		A	HYPERSTORE	REDIS MON	REDIS CRED	REDIS QOS	S3
hyperstore-demo1	✓	✓	✓		✓ •	✓	✓
hyperstore-demo3	✓	✓	✓		✓	✓ •	✓
hyperstore-demo5	✓	✓	✓	✓ •			✓
hyperstore-demo1b	✓	✓	✓				✓

VIEW NODE STATUS

For each datacenter, each Cloudian HyperStore node in the datacenter is represented by a cube. Clicking on the cube, allows you to view individual node activity and manage node services. Clicking on Host: allows individual node details, including sections on disk detail, service status and node specific alerts.



Host: DC1 RAC1 hyperstore-demo1

DISK READS PER SECOND
6.4 MB

DISK WRITES PER SECOND
6.4 MB

NETWORK TRAFFIC:
RECEIVED: 6.3 MB/SEC
TRANSMITTED: 6.4 MB/SEC

REQUEST THROUGHPUT:
GET: 564
PUT: 420

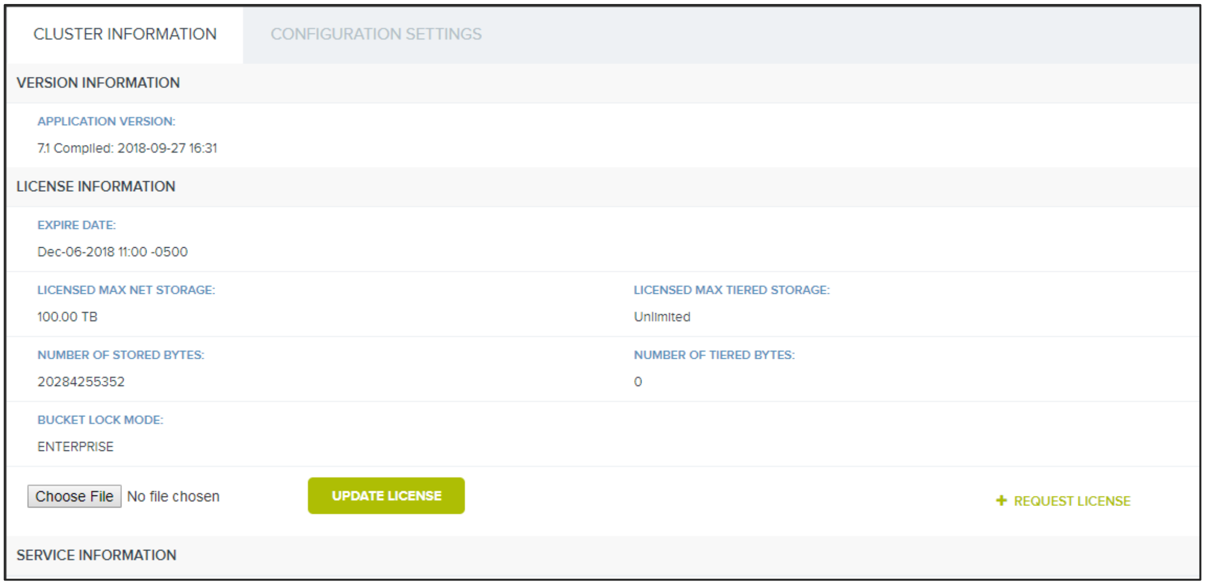
TRANSACTIONS/SEC:
GET: 0.0
PUT: 0.0

AVERAGE REQUEST LATENCY (MS):
GET: 20
PUT: 151

STATUS	DEVICE	MOUNT POINT	USE TYPE	DISK USAGE
	/dev/mapper/vg0-root	/	CASSANDRA, REDIS, LOG	28.3 GB of 180.7 GB used
	/dev/sdb1	/cloudian1	HS	629.1 GB of 3.9 TB used
	/dev/sdc1	/cloudian2	HS	918.8 GB of 3.9 TB used
	/dev/sdd1	/cloudian3	HS	426.3 GB of 3.9 TB used
	/dev/sde1	/cloudian4	HS	553.3 GB of 3.9 TB used
	/dev/sdf1	/cloudian5	HS	876.7 GB of 3.9 TB used
	/dev/sdg1	/cloudian6	HS	841.7 GB of 3.9 TB used

SIMPLE CLUSTER SETTINGS

The Cluster Information page displays static information about your Cloudian HyperStore system such as software version, service hosts and license info. Note this cluster is enabled for WORM as the bucket lock mode is set to ENTERPRISE.



VERSION INFORMATION

APPLICATION VERSION:
7.1 Compiled: 2018-09-27 16:31

LICENSE INFORMATION

EXPIRE DATE:
Dec-06-2018 11:00 -0500

LICENSED MAX NET STORAGE:
100.00 TB

LICENSED MAX TIERED STORAGE:
Unlimited

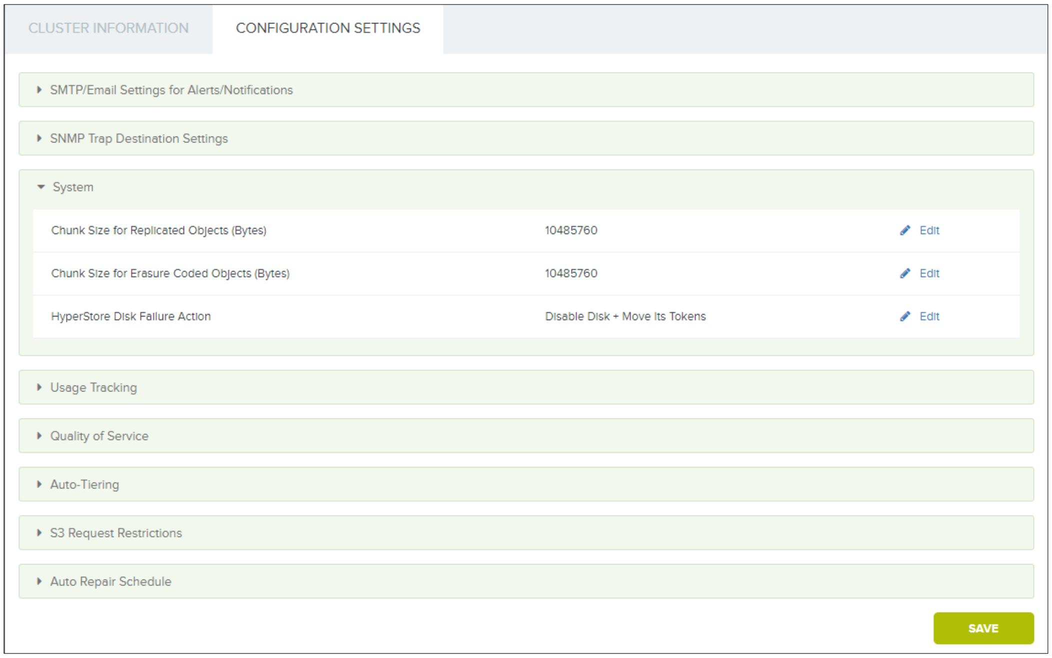
NUMBER OF STORED BYTES:
20284255352

NUMBER OF TIERED BYTES:
0

BUCKET LOCK MODE:
ENTERPRISE

SERVICE INFORMATION

The configuration settings page allows for globally modifying and updating the configuration files, requiring no service restart.


CLUSTER INFORMATION
CONFIGURATION SETTINGS
SMTP/Email Settings for Alerts/Notifications
SNMP Trap Destination Settings
System

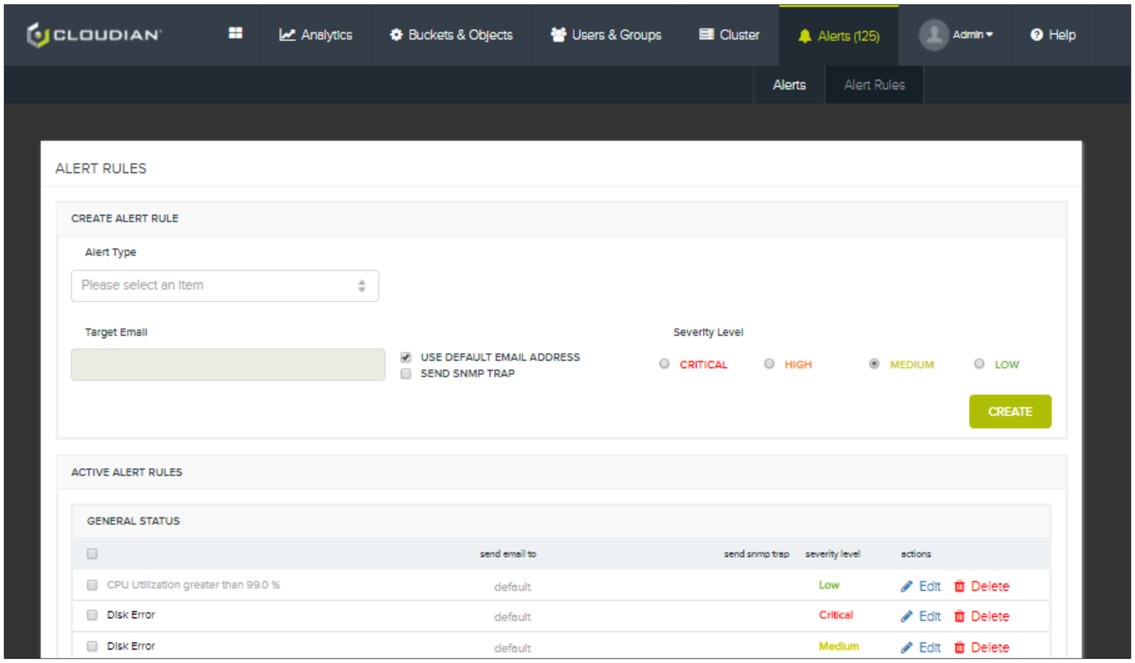
Chunk Size for Replicated Objects (Bytes) 10485760 Edit

Chunk Size for Erasure Coded Objects (Bytes) 10485760 Edit

HyperStore Disk Failure Action Disable Disk + Move Its Tokens Edit

SIMPLE NOTFICATIONS & ALERTS

The Cloudian HyperStore system comes with a set of pre-configured notification rules. The pre-configured notification rules are listed in the “Rules” section of the “Notification Rules” page.



ALERT RULES

CREATE ALERT RULE

Alert Type: Please select an item

Target Email

☑ USE DEFAULT EMAIL ADDRESS
☐ SEND SNMP TRAP

Severity Level: ○ CRITICAL ○ HIGH ● MEDIUM ○ LOW

CREATE

ACTIVE ALERT RULES

GENERAL STATUS

☐	send email to	send snmp trap	severity level	actions
☐ CPU Utilization greater than 99.0 %	default		Low	Edit Delete
☐ Disk Error	default		Critical	Edit Delete
☐ Disk Error	default		Medium	Edit Delete

PERFORMANCE

This section discusses various performance topics. Following the guidelines in this section will help ensure that optimal HyperStore performance is realized.

PERFORMANCE ENVRIONMENTAL FACTORS

Before installing or adding new HyperStore nodes, it is important to consider environmental factors. The first important factor is that one or more load balancers are utilized when using Cloudian Hyperstore in a production environment. Load balancers will optimize the S3 network traffic flow by ensuring that all nodes in the cluster are equally accessed with read and write requests. There is one exception. Since the Cloudian Management Console (CMC) does not share its active session with other nodes in the cluster, exclude this node from the load balancing configuration.

Other environment factors are critical as well, including number of network ports per node, overall network design, physical switch and port configuration settings. All these factors should be considered during the assessment and planning phases to ensure the environment is ready and optimized for the HyperStore cluster deployment.

Once HyperStore is deployed, it is important to do a storage performance baseline with one or more I/O workloads and make any needed environmental changes to ensure the best performance possible. Note that further performance tuning may be necessary depending on specific HyperStore applications and the I/O workloads that results from your specific application. Finally, prior to going live, do a proof-of-concept (POC) to ensure your application has met the desired performance criteria based on earlier baseline and POC results.

PERFORMANCE OPTIMZATION SCRIPTS

The HyperStore system includes a performance configuration optimization script that is automatically run on each node when you install HyperStore. This script also automatically runs on any new nodes that you subsequently add to your cluster. The script updates OS configuration settings on each node and a few critical HyperStore system configuration settings for optimal performance of the cluster. This script adjusts the settings based on the hardware environment in which it is executed by considering factors such as RAM resources and CPU specs. After automatically running, the cluster performance will scale in a predictable way as more nodes are added to the cluster.

Since the script runs automatically during installation and when adding nodes during cluster expansion, you should not need to run the script yourself. There is an option where you can run the performance configuration optimization script indirectly through the installer's Advanced Configuration Options menu. This is a rare event and should only be done if you have made configuration changes and your system is now under-performing as a result. Running the script on one or more nodes will return the configuration settings to the optimized values. Once the script is run, again performance scales in a predictable way as more nodes are added to the cluster. This script is continually updated with each new release of HyperStore, ensuring that performance continue as new generations of hardware become available.

CONCLUSION

Cloudian HyperStore software makes it easy to build fully-featured, Amazon S3-compliant cloud storage, on-premises. It is available as either stand-alone software, or as Cloudian HyperStore appliances. Either way, Cloudian HyperStore software ensures unlimited scale, multi-datacenter storage, fully automated data tiering, and support for all S3 applications—all behind your firewall.

Cloudian HyperStore software, whether deployed on a user's existing hardware or pre-installed on a Cloudian HyperStore appliance, combines robust availability with system management control, monitoring capabilities and reporting. A host of features, including hybrid cloud streaming, virtual nodes, configurable erasure coding, and data compression and encryption sets Cloudian apart with highly efficient storage and seamless data management that lets users store and access their data where they want it, when they want it. Built on a robust object storage platform for effortless data sharing, cloud service providers around the world use Cloudian HyperStore to deploy and manage storage for both public and private clouds, while enterprises rely on it to store their backups, media and entertainment data, medical images and for 24-hour video surveillance data both in their private and hybrid clouds.

CLOUDIAN®

CLOUDIAN, INC.
177 Bovet Road, Suite 450, San Mateo, CA 94402
Tel: 1.650.227.2380 | cloudian.com