

Clouddian HyperStore[®] Technical Whitepaper



CONTENTS

INTRODUCTION	1
CLOUDIAN HYPERSTORE OVERVIEW	2
Cloudian HyperStore Features	3
Scale as You Grow	3
Protects all Data	3
Effortless Data Movement	3
100% S3 Compatible	4
Security	4
Encryption under Cover	4
Multi-Datacenter and Multi-Region	4
Self-Service IT	4
Multi-Tenancy	5
Integrated Billing, Management and Monitoring	5
Broad Application Support	5
SMB/NFS/FTP Support	5
Cloudian HyperStore Secret Sauce	5
Distributed Peer-to-Peer Architecture	6
Virtual Node	6
Parallel Disk I/O Data Protection	6
Configurable Data Consistency	6
Storage Node Heterogeneity	6
Compression Your Way	7
Quality of Service (QoS)	7
Cloud Storage File System (CSFS)	7
CLOUDIAN HYPERSTORE ARCHITECTURE	7
Cloudian Hyperstore Geocluster	8
Regions	9
Cloudian Hyperstore Components	9
Services	10



CLOUDIAN HYPERSTORE OPERATIONS	13
User Management	13
Data Protection	13
Supported Erasure Coding Configurations	14
Datacenter Assignment	15
Smart Repair	15
Auto Tiering	15
Quality of Service (qoS)	16
Storing Object Metadata	16
CLOUDIAN HYPERSTORE INTERNALS	17
Object Metadata	17
Cloudian HyperStore vNodes	17
How vNodes Work	18
Server Side Encryption	20
ACCESS PROTOCOLS AND APPLICATIONS	20
100% S3 Compatible Interface	20
Advance S3 Compatibility	21
Extensions to the S3 API	22
Bucket Storage Type Configuration	22
Virtual Buckets	22
Canned ACL group-read	23
Return User-Defined Object Metadata	23
Transition Rules	23
User Management	23
Group Management	23
Public URL Service	23
QoS Limits Service	23
Usage Reporting	24
Rating Plan Service	24
Billing Service	24
Billing Whitelist Service	24
System Services	24
Developing S3 Applications	24

STORING FILES IN CLOUDIAN HYPERSTORE	25
Hyperstore Connect for Files	25
Deployment Models	26
Kernel File System	26
High Performance I/O	26
Dynamic Real-Time Snapshot, Version Control and Global Multi-User Accessibility	27
Global File System	27
Sub-File Block, Global, Pre-Move Deduplication	27
Global Cloud Deduplication	28
Data Compression	28
File System Cache	29
Audit and Repair, True File Immutability, Failure Probability and Prediction	29
Global View Management Console and Locality Aware Optimization	30
Unlimited Performance Scalability, Identity Inheritance and Direct Accessibility	30
In Front of and Behind of Automated Tier Management	31
CLOUDIAN HYPERSTORE MANAGEMENT	32
One Simple Web-Based GUI	32
Configurable	32
Programmable	32
View Summary for Details	33
Simple Dashboard	33
Single Cluster Usage & Performance View	33
Capacity Explorer	34
View User and Tenant Usage	34
Storage Policies	35
Cluster Configuration & Monitoring	36
Multi-Datacenter & Region View	36
View Node Status	36
Simple Cluster Settings	37
Simple Notifications & Alerts	37
CONCLUSION	38
ABOUT CLOUDIAN	38

INTRODUCTION

With the popularity of rich media, the proliferation of mobile devices and the digitization of content, there has been an exponential growth in the amount of unstructured data that IT is managing—and this growth is not slowing but instead accelerating. This unprecedented growth in unstructured content is simply not sustainable for current NAS/SAN infrastructures. In fact, the whole storage system is breaking down. Backups and restores are taking longer. Migrations from older storage systems to new storage systems are labor intensive. Provisioning storage for users is more frequent and time consuming, and the list goes on and on.

Not only does all of 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. A drastic change in storage infrastructure is needed if enterprise IT is ever going to tame the data explosion.

Software-defined object storage offers an alternative approach to NAS/SAN systems. It gives enterprises the ability to leverage the latest advancements in cost-effective commodity CPU and storage technology. Allowing enterprises to keep their environments in lock-step with the ever increasing storage and IO demands of critical business applications. For software-defined storage (SDS) architectures more powerful CPUs lead to greater scale and performance. Intel, for example, typically comes out with a new CPU product every 6-12 months. Compare this with the typical 3-year product refresh cycle from proprietary storage vendors and 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). SDS architectures allows enterprises to take advantage of these technology updates earlier, gaining significant efficiency benefits.

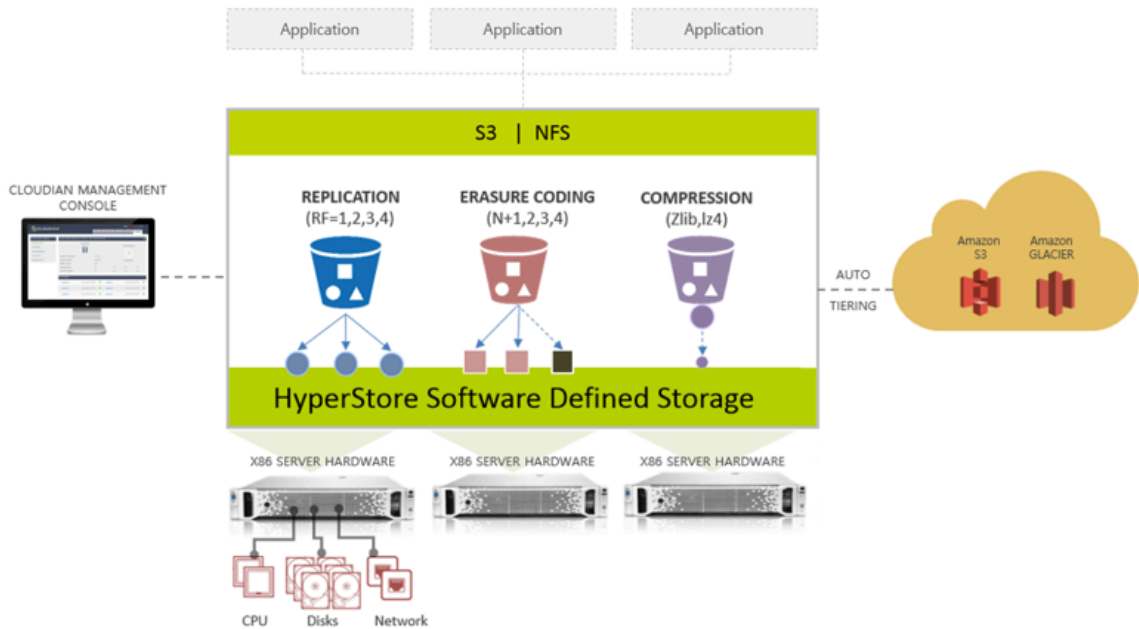
Cloudian HyperStore® is a scale-out object storage system designed to manage massive amounts of data. It is an SDS platform which runs on any standard Linux servers – this dramatically reduces the cost for datacenter storage while still providing extreme availability and reliability.

In this whitepaper, we will provide an in-depth review of Cloudian HyperStore, how it functions, a technical implementation perspective and the unique capabilities of the product.















CLOUDIAN HYPERSTORE OVERVIEW

Software-defined hybrid cloud storage solutions, such as Cloudian HyperStore, enable datacenters to create highly cost-effective unstructured data storage repositories built on standard hardware that spans across the enterprise as well as out into public cloud environments.



Cloudian HyperStore makes it easy to build full-featured, Amazon S3-compliant cloud storage, on-premises. It is available as a stand-alone software or fully integrated with 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.

HyperStore Software
Enterprise-Scale. Unprecedented Efficiency.

 Natively S3	 Hybrid Storage Cloud	 Extreme Durability	 Multi-Tenant
 Geo-Distribution	 Intelligence in Software	 Scale out	 Smart Support
 Billing & Reporting	 Data Protection	 Quality of Service	 Programmable

CLOUDIAN HYPERSTORE FEATURES

Cloudian HyperStore software, whether deployed on existing hardware or pre-installed on Cloudian HyperStore appliances, 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 as well as encryption helps set Cloudian HyperStore 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 both public and private clouds, while enterprises rely on it to maintain their private and hybrid clouds.

SCALE AS YOU GROW

Cloudian HyperStore cost-effectively scales via standard hardware so initial deployments can be comprised of only a few servers in a single datacenter. As demand grows, these installations can easily expand to thousands of servers that process hundreds of petabytes of data, distributed across multiple datacenters.

PROTECTS ALL DATA

Cloudian HyperStore software optimizes storage for all data objects, both large and small. Erasure coding enables deep archive efficiency and flexible redundancy, giving you robust data protection without consuming precious disk space. Object replicas are employed for frequently used data.

EFFORTLESS DATA MOVEMENT

Cloudian HyperStore affords you the flexibility to put your data where you want it and easily retrieve it when you want it. With unique features like object streaming and dynamic auto-tiering, data moves seamlessly between your on-premises cloud and Amazon S3, regardless of file size.

100% S3 COMPATIBLE

Several years ago, public cloud providers realized that applications are driving the need for cloud storage. Amazon Web Services (AWS) started a cloud revolution in early 2006 when they launched the Amazon Simple Storage Service (Amazon S3). Amazon S3 is a massively scalable, cost-effective cloud storage solution developed specifically to house the massive influx of data created by organizations worldwide. Today Amazon S3, the public cloud giant, has achieved several milestones including the number of objects stored in S3, now more than 2 trillion and the development of an expanding ecosystem of over 350+ compatible applications. It's clear that Amazon has established itself as the dominant leader in public cloud storage. Now, emerging technologies like hybrid cloud storage are starting to receive attention from cloud providers and enterprise IT. For the enterprise, flexibility, control, and resilience are at the forefront of their storage infrastructure concerns. Hybrid cloud is rapidly being adopted as the solution of choice that addresses storage requirements for enterprises.

SECURITY

With data security breaches becoming more commonplace, it is essential for businesses to safeguard their data from the prying eyes of data hackers and unauthorized users. Cloudian HyperStore simplifies the data encryption process by providing transparent key management at the server or node layer. This relieves administrators from the burden of having to manage encryption keys and eliminates the risk of data loss occurring due to lost keys. Furthermore, encryption can be managed very granularly—either at a bucket level or down to an individual object.

Cloudian HyperStore AES-256 Server-Side Encryption enables enterprises and service providers to easily encrypt data stored at rest. SSL encryption ensures data confidentiality for data in transit (HTTPS). And with S3-compatible object-level ACLs, system administrators can better manage access to buckets and objects.

MULTI-DATACENTER AND MULTI-REGION

Cloudian HyperStore software can further scale while increasing durability by replicating data across multiple datacenters (availability zones). Reads and writes are always performed at the local datacenter with remote replication performed in the background to avoid latency of remote writes. Cloudian HyperStore software supports multiple regions with shared multi-tenant management to enable both geographic optimization and unlimited scalability. Data can be placed in specific regions for security, policy, cost or other reasons.

SELF-SERVICE IT

Robust management controls, including system health monitoring, make non-disruptive system maintenance and up-grades a snap. As a multi-tenant solution, Cloudian HyperStore software gives role-based access to system and group administrators and to users. Cloudian HyperStore provides an easy to use GUI and REST API to facilitate quotas, user provisioning, system administration, usage reporting, billing automation at the group level, and flexible rating plans. Quality of Service (QoS) controls allow IT administrators to control access and usage.

MULTI-TENANCY

Advanced identity and access management 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. Both administrators and users benefit from unique reporting operations and account and data management capabilities. Multiple credentials per user is also supported. Configurable group—and user-level QoS rate limits ensure groups and users do not exceed storage quotas or consume bandwidth in a manner that impacts other tenants.

INTEGRATED BILLING, MANAGEMENT AND MONITORING

Cloudian HyperStore provides system/cluster monitoring and data management, full provisioning and management of users, groups, rating plans, QoS controls and billing via graphical user interface or REST-ful APIs. This comprehensive API coverage enables tight integration with the user's provisioning, authentication and billing systems. Support for multiple billing and subscription revenue models flexibly supports both cloud service providers (CSPs) and enterprise chargebacks. The graphical user interface is highly and easily customizable in order to provide better integration in existing environments.

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.

CLOUDIAN HYPERSTORE SECRET SAUCE

Built on Cassandra no-SQL database, enhanced with file system properties, 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 with no reliance on any special hardware for PB scale durability, availability & storage



Distributed Everything

All data, metadata, 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 ARCHITECTURE

Cloudian storage clouds are implemented by deploying individual nodes comprised of CPUs and disk drives into a logical “ring” architecture. As physical nodes are added, all the resources are aggregated into a common pool of storage and CPU resources across the ring. Rings can start out with as little as one node, however, for redundancy and availability purposes, two or three nodes are typically deployed in an initial implementation. Single nodes can then be added to the ring as needed. Data is dispersed across the ring, 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 (vNodes). This allows for greater IO parallelism and hence greater storage IO performance across the ring. Secondly, it enhances availability because 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 storage devices are added to the ring, the higher the likelihood that a drive will fail. To compound this problem, disk manufacturers are now coming out with ultra-high-density 10 terabyte (TB) disk drives. The RAID rebuild times for recovering these drives 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 in combination with a massively parallel disk IO architecture, Cloudian HyperStore dramatically shrinks drive rebuild times and decreases the window of exposure during the repair process.

CONFIGURABLE DATA CONSISTENCY

Cloudian HyperStore also provides the ability to configure the level of data consistency when using replication to protect objects across a storage ring. For example, the default consistency requirement for read and write operations is defined as “quorum”, meaning that a read or write operation must succeed on a quorum (or set number) of replica copies before a success response is returned to the client application. This enables users to configure how stringent they wish their replication policy to be. For example, for those data objects that are considered mission critical, the replication policy is set to wait until an acknowledgment is received from nodes across multiple datacenter locations before an acknowledgment is sent back to the application. On the other hand, 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 ring can be of dissimilar size. 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 ring. This gives businesses the flexibility to add capacity and CPU resources as granularly as desired. It also helps to improve efficiencies as the right resources can be added to the ring at the right time.

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 less data to store 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.

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. A Chief Financial Officer (CFO) could be assigned a high priority privilege (e.g. Platinum Service Level) to financial records while an end-user accessing the data could be given lower priority access (e.g. Silver Service Level). In short, QoS and metering are foundational capabilities for implementing a multi-tenant private cloud storage solution.

CLOUDIAN HYPERSTORE ARCHITECTURE

Cloudian HyperStore is an Amazon S3-compliant multi-tenant object storage system. The system utilizes a not only 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:

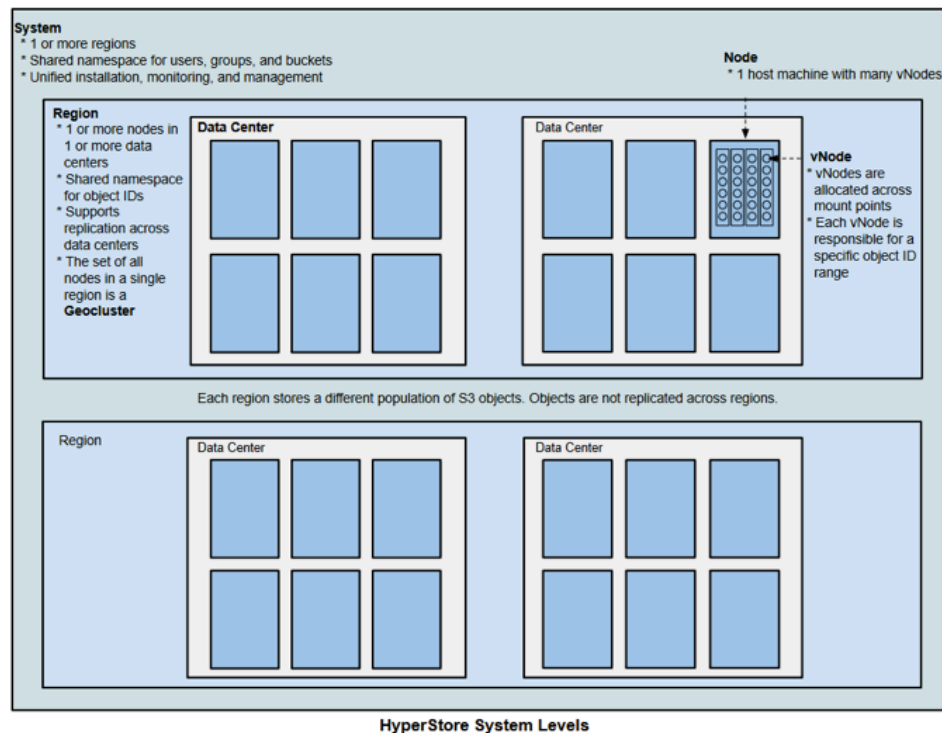
- **Amazon S3 API compliance.** The Cloudian HyperStore system is fully compatible with Amazon S3's HTTP REST API. Customers' existing HTTP S3 applications will work with the Cloudian HyperStore service, and existing S3 development tools and libraries can be used for building Cloudian HyperStore client applications.
- **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 and 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. Objects can also be exposed via public URLs for regular web access, subject to configurable expiration periods.
- **Reporting and billing.** The Clouidian HyperStore system supports usage reporting on a system-wide, group-wide, or individual user basis. Billing of groups or users can be based on storage quotas and usage rates (such as bytes in and bytes out).
- **Horizontal scalability.** Running on standard off-the-shelf hardware, a Clouidian 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 Clouidian 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 Clouidian HyperStore geo-cluster can be deployed across multiple datacenters to provide redundancy and resilience in the event of a datacenter scale disaster.

The following diagram presents the major components of the Clouidian HyperStore architecture: Clouidian Management Console (CMC), S3 Service, Administrative Service, Clouidian HyperStore storage services, Redis and Cassandra databases.

CLOUDIAN HYPERSTORE GEO-CLUSTER

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



REGIONS

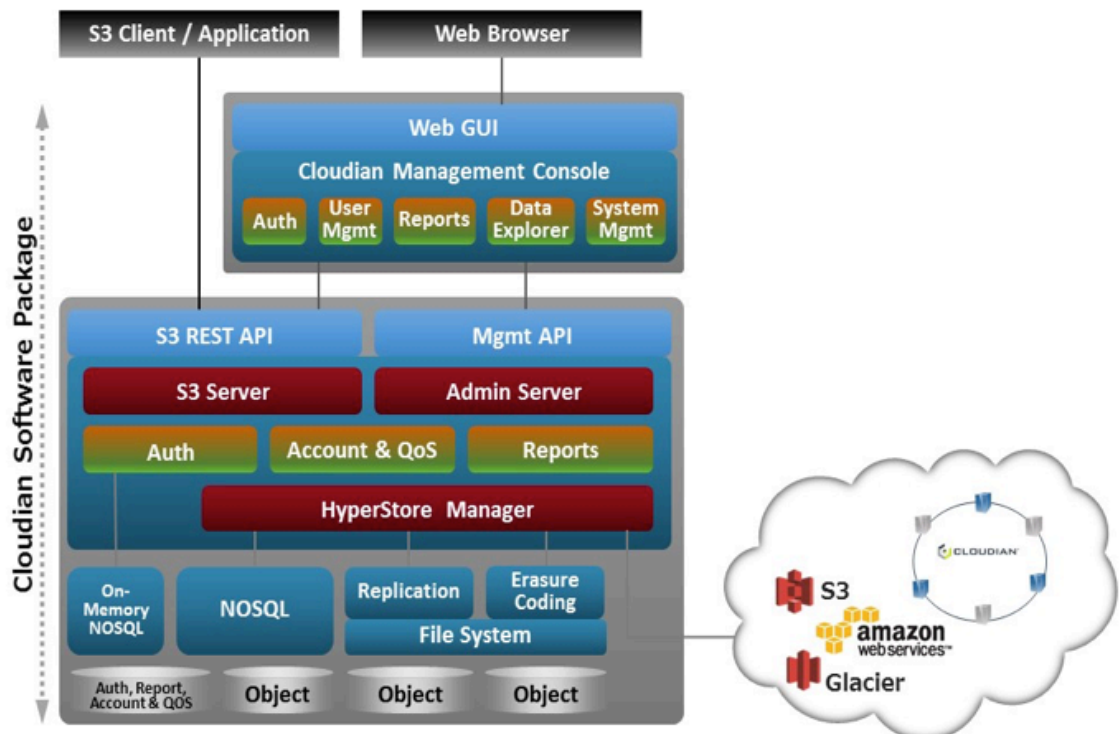
Like Amazon S3, the Cloudian HyperStore system supports the implementation of multiple "service regions". Setting up the Cloudian HyperStore system to use multiple service regions is optional.

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 service offering (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.

CLOUDIAN HYPERSTORE COMPONENTS

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



SERVICES

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

SERVICE NAME	DESCRIPTION AND ROLE
S3 Service	<p>The Clodian HyperStore system provides a high-performance S3 proxy service. The S3 Service processes S3 REST requests incoming from client applications</p>
Clodian HyperStore Service and the HSFS	<p>As an object store, Cassandra provides a wealth of valuable built-in functionality including data partitioning, automatic replication, easy cluster expansion, quorum calculation, and so on.</p> <p>The Clodian 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 Clodian HyperStore File System (HSFS).</p> <p>In Clodian 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 on and HSFS is used as the storage layer to store S3 object data.</p> <p>Within the HSFS, objects can be stored and protected in either of two ways:</p> <ul style="list-style-type: none"> • Replicated storage • Erasure coded storage
Cassandra DB Services	<p>The Clodian HyperStore system uses the open source storage platform Cassandra to store several types of data. The Clodian HyperStore system creates and uses several distinct "key spaces" within Cassandra.</p> <p>S3 client applications do not access Cassandra databases directly; all S3 access is mediated through the S3 Service.</p>

Redis DB Services

The Clodian HyperStore system uses the lightweight, open source Redis key-value data store to store a variety of data that supports Clodian HyperStore S3 service features. There are two types of Redis databases (DBs) in a Clodian 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 Clodian HyperStore Service are the clients to these two Redis DBs. Communication is through a protocol called Redis Serialization Protocol (RESP).

Administrative Service

The Clodian 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 CMC is a client to the Administrative Service. You also have the option of building your own Administrative Service client.

The Administrative Service is closely integrated with the S3 Service. Both leveraging Jetty technology; both are installed together; and both are started and stopped together by the same commands.

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.

Group administrators can perform a more limited range of administrative tasks pertaining to their own group. Regular users can perform S3 operations such as 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 a lightweight domain resolution utility suitable for small networks. This utility is bundled with Cloudian HyperStore software. The Cloudian HyperStore interactive installation wizard automatically installs and configures dnsmasq to resolve Cloudian HyperStore service domains (specifically the S3 service domain, the S3 website endpoint domain, and the CMC domain) based on user input provided during the installation. If you don't want to install and use dnsmasq, when you launch the install script you can use a no-dnsmasq option.

CLOUDIAN HYPERSTORE OPERATIONS

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

- User Management
- Data Protection
- Multi Datacenter Distribution
- Repair
- Auto-tiering
- Quality of Service (QoS)

USER MANAGEMENT

Through the Cloudian HyperStore Administrative API or through the CMC, you can provision the user groups and individual users who you want to authorize to use the Cloudian HyperStore S3 service. The Cloudian HyperStore system can support millions of groups and users. You will provision groups 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 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 of the group as a whole. Each group can also be assigned default user-level QoS controls that will limit the service usage of individual users within the group. (Optionally, you can also assign per-user QoS limits that will supersede this 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. (Optionally, you can also assign per-user rating plans that will supersede this default.)

DATA PROTECTION

Central to Cloudian's data protection are 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 when they create a new storage bucket can then choose which pre-configured storage policy to use to protect data in that bucket. 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.

- **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.

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 that are infrequently accessed.

Regardless of whether you use replication or erasure coding, if your Cloudian HyperStore system spans multiple datacenters, for each storage policy you can also choose how data is allocated across your datacenters—for example, you could have a storage policy that for each S3 object stores 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 three erasure coding configurations:

- 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.
- 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.
- 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.

The choice among these three supported EC configurations is largely a matter of how many Cloudian HyperStore nodes you have in the datacenter. 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.

DATACENTER ASSIGNMENT

If your Clouidian HyperStore system is deployed across multiple datacenters, for each storage policy that you create you can configure a datacenter assignment scheme for the policy. This determines which of your datacenters to use for storing data, for each storage policy.

You have the option of creating a storage policy that stores data in some of your datacenters and not others—for example, store data in DC1 and DC2 but not in DC3. Note, however, that DC3 may be involved in processing S3 requests associated with buckets that use this policy. There is only one S3 service endpoint per region, and incoming S3 requests may resolve to any DC within the region. If the S3 Service in DC3 receives an S3 PUT request in association with a policy that stores data only in DC1 and DC2, it will transmit the uploaded object on to DC1 and D2 (it will not be stored in DC3). Likewise, if DC3 receives an S3 GET request in association with a policy that stores data only in DC1 and DC2, then DC3's S3 Service will get the object from DC1 or DC2 and pass it on to the client. If you want more absolute barriers so that for example DC3 never touches DC2's data and vice-versa, you need to set up your system so those DCs are in different service regions.

SMART REPAIR

It's important that replicated data in your Clouidian HyperStore storage cluster be assessed and repaired regularly to ensure that for each data object the proper number of replicas exist across the cluster, and that all replicas are up-to-date. If you use erasure coding (EC) in your Clouidian HyperStore system, it's also important that erasure coded data be regularly evaluated and repaired to ensure that for each object the right number of fragments exist across the cluster and that all fragments are up-to-date.

Toward this end, the Clouidian HyperStore system implements a repair-on-read feature. When a read request is processed for a particular replicated object in the Clouidian 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, for a percentage of reads (applied randomly to about 20% of EC object reads and about 10% of Cassandra data reads).

However, this mechanism repairs only data that is read. It's necessary therefore to have also have a mechanism that regularly checks all data in the system and implement repairs as needed.

Regular repair of all data is accomplished by the Clouidian HyperStore auto-repair feature which automatically executes repair jobs on each node in your cluster on a configurable schedule. Separate repair jobs are run for Cassandra data (object metadata and service usage data that's replicated across the cluster and stored in Cassandra), replica data (S3 object data that's replicated across the cluster and stored in the Clouidian HyperStore File System [HSFS]), and EC data (erasure coded S3 object fragments that are distributed across the cluster and stored in the HSFS).

AUTO TIERING

The Clouidian HyperStore system supports an "auto-tiering" feature whereby objects can be automatically moved from local Clouidian HyperStore storage to a destination storage system on a

predefined schedule. The destination storage system can be any of the following:

- Amazon S3
- Amazon Glacier
- A different Cloudian HyperStore region or system

Auto-tiering is configurable on a per-bucket basis. By default, auto-tiering is not enabled for any bucket, and the CMC functionality for activating auto-tiering is hidden. If you want bucket owners to be able to activate and configure auto-tiering for their buckets, you will need to configure the CMC so that it displays this functionality for CMC users.

QUALITY OF SERVICE (QOS)

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

- 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.

The Cloudian HyperStore system enforces QoS settings by rejecting 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.

STORING OBJECT METADATA

Like Amazon S3, Cloudian HyperStore allows for rich metadata to be associated with each stored object. The system allows for user-defined object metadata as well as system-defined object metadata.

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 Cloudian HyperStore system 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.

In regard to storage policy, Cloudian HyperStore provides system operators the option to configure different consistency requirements for replicated object metadata than for the replicated S3 object data itself.

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.

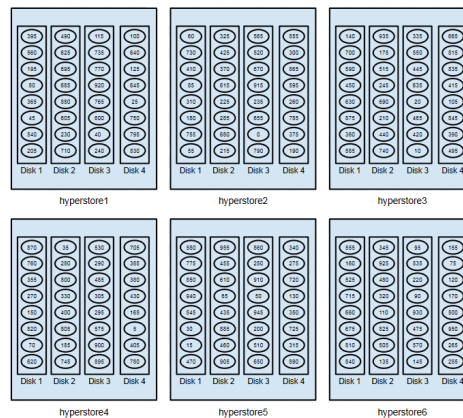
HOW VNODES WORK

Traditionally, consistent hashing 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 "virtual nodes", with multiple virtual nodes residing on each physical host.

The Cloudean HyperStore system goes a significant step further by assigning a different set of tokens (virtual nodes) to each disk on each physical host. With this implementation, 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. The other disks on the host can continue operating and supporting their own data storage responsibilities.

For illustration, consider a geo-cluster of six Cloudean HyperStore hosts each of which has four disks designated for S3 object storage. Suppose that each physical host is assigned 32 tokens. And suppose for illustration 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 diagram below 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.

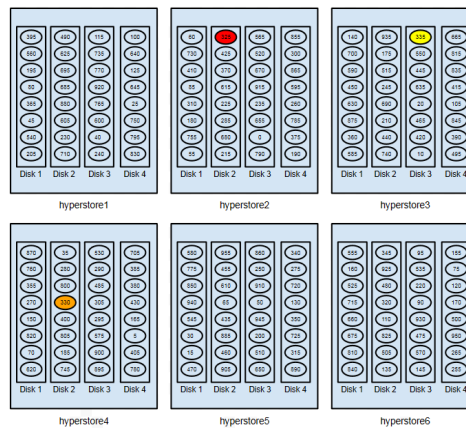


Now further suppose that you've configured your Cloudean HyperStore system for 3X replication of S3 objects. And say that an S3 object is uploaded to the system and the hashing algorithm applied to the object name gives us a hash value of 322 (in this simplified hash space). The diagram below shows how 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—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 Cloudean HyperStore2:Disk2, so that's

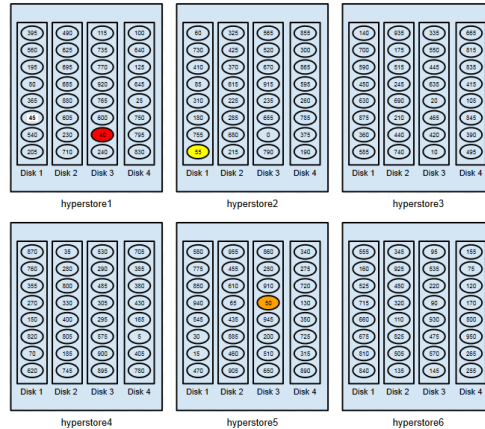
where the primary replica of the object is stored. Note that the “primary replica” has no functional primacy compared to other replicas; it’s called that only because its placement is based simply on identifying the disk that’s assigned the token range into which the object hash falls.

- 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 Cloudian HyperStore vNode principle: no more than one of an object’s replicas will be stored on the same physical host. Suppose that 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 Cloudfian HyperStore system supports server-side encryption (SSE) to protect the confidentiality of data at rest. And like Amazon, the Cloudfian HyperStore system can perform the encryption (and subsequent decryption upon object retrieval) either with a system-generated encryption key (regular SSE) or a customer-provided encryption key (SSE-C).

For both encryption approaches, the Cloudfian HyperStore system works just like Amazon S3. This includes that for encryption with a customer-provided encryption key:

- 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 encryption key. If an object is uploaded to Cloudfian 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.

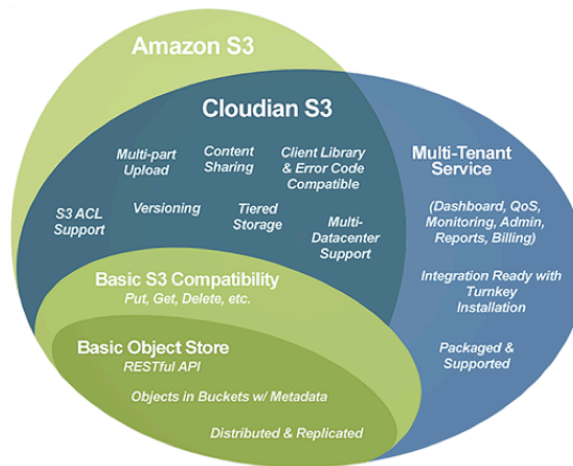
ACCESS PROTOCOLS AND APPLICATIONS

Cloudfian 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 COMPATIBLE INTERFACE

Cloudfian is the only storage platform that offers 100% advanced 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 this advanced functionality, Cloudian is the most compatible storage platform for S3 on-premises and hybrid cloud deployments.



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 its 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 S3 and the private cloud, it is of utmost importance to understand the level of compatibility a storage platform claims versus its compatibility in reality. S3 is quickly becoming the object storage standard and will not disappear anytime soon. Readers of this document are likely a current user or plan to use it, either exclusively or in a hybrid cloud environment. 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 51 total operations available through it. 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 51 operations available through the S3 API, 24 of them are considered advanced. Figure 3 below, depicts simple operations [illustrated in red], moderately complex operations [illustrated in yellow] and advanced operations [illustrated in green]. To be considered compatible to an advanced degree, a storage platform should be able to perform the majority of the advanced operation

Operations on the Service	Operations on the Buckets	Operations on the Objects	Additional Features
Get	Get Bucket (List Objects)	Get Object	Initiate Multipart Upload
	Put Bucket	Put Object	Upload Part
	Delete Bucket	Delete Object	Upload Part - Copy
	Head Bucket	Head Object	Complete Multipart Upload
	Get Bucket Website	Get Object ACL	Abort Multipart Upload
	Put Bucket ACL	Put Object ACL	List Part
	Put Bucket Website	Put Object (Copy)	
	Delete Bucket Website	Delete Multiple Objects	
	List Multipart Uploads	Options Object	
	Get Bucket ACL	Post Object	
	Get Bucket Lifecycle	Get Object Torrent	
	Get Bucket Policy	Post Object Restore	
	Get Bucket Tagging		
	Put Bucket Lifecycle		
	Delete Bucket Lifecycle		
	Delete Bucket Policy		
	Get Bucket CORS		
	Get Bucket Location		
	Get Bucket Logging		
	Get Bucket Notification		
	Get Bucket requestPayment		
	Get Bucket Versioning		
	Put Bucket Policy		
	Get Bucket Object Versions		
	Put Bucket CORS		
	Put Bucket Versioning		
	Put Bucket Logging		
	Put Bucket Notification		
	Put Bucket Tagging		
	Put Bucket requestPayment		
	Delete Bucket CORS		
	Delete Bucket Tagging		

EXTENSIONS TO THE S3 API

Cloudian HyperStore offers 100% S3 compatibility including advanced features and also extends the S3 API to gain additional functionality. This chapter will explain each of the Cloudian HyperStore 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.

BUCKET STORAGE TYPE CONFIGURATION

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 are storing and replicating such objects in the Cloudian HyperStore File System, 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 by the request-receiving S3 Server to the correct region. After request processing, the request-receiving S3 Server will return a response to the client. Regardless of which region a virtualized bucket's objects are stored, all usage data for the bucket (for purposes of QoS enforcement and billing) is tracked at the region in which the bucket was created.

CANNED ACL GROUP-READ

Canned ACL Group-read allows read access to everyone in the object owner's group, to allow permissions to a group other than the object owner's group. 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, and in a third party Cloudian HyperStore service.

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 allows 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:

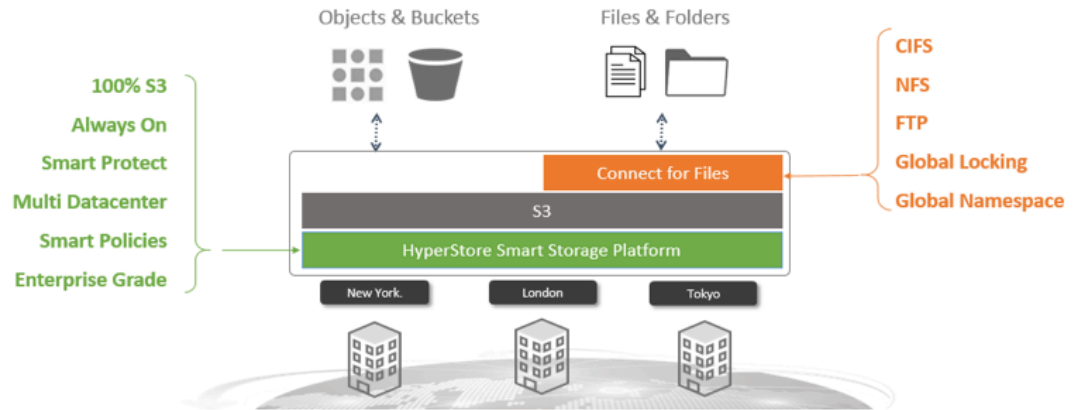
- [AWS Java Developer Center](#)
- [AWS Windows/.NET Developer Center](#)
- [AWS PHP Developer Center](#)
- [AWS Python Developer Center](#)
- [AWS Ruby Developer Center](#)

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

STORING FILES IN CLOUDIAN HYPERSTORE

Cloudian HyperStore Connect for Files allows enterprises to offer scalable file services on top of Cloudian HyperStore object storage using industry standard protocols like NFS, CIFS & FTP. File based environments can gain from the extreme durability, availability, geo dispersal, multi-tenancy and low cost of Cloudian HyperStore object storage.



HYPERSTORE CONNECT FOR FILES

Cloudian now delivers a global namespace with standard file access protocols. It provides global file services on top of Cloudian HyperStore object storage system. Cloudian HyperStore Connect for Files is comprised of two modules – Access Points and optional Global View Manager.

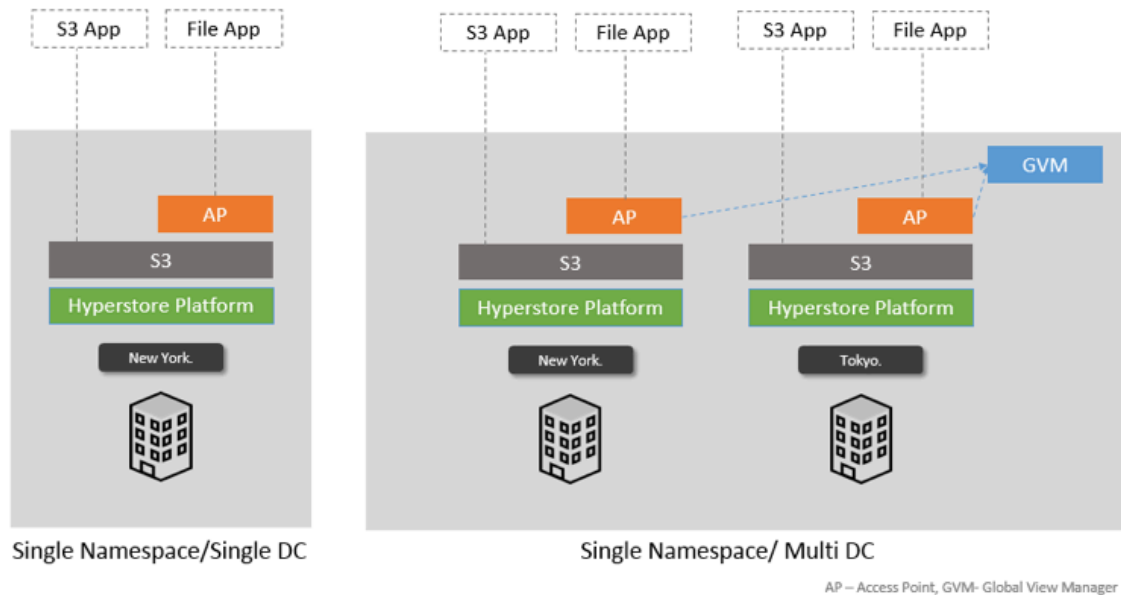
File Access Point (FAP)

Access Points provide an easy and secure method for providing file access to data centers, workgroups and remote locations just like traditional file servers. Access Points translate file protocols such as SMB, NFS, and FTP directly to Cloudian HyperStore object storage using S3 protocol. Access Points can be installed as virtual or physical appliances with secure local file access.

Global View Manager (GVM)

Global View Manager designed for organizations requiring global file accessibility, collaboration and workflow distribution across geographically dispersed offices. Customers connect multiple access points to a Global View Manager to offer a Global namespace or view. It also ensures namespace consistency and file locking between sites.

DEPLOYMENT MODELS



KERNEL FILE SYSTEM

CSFS has been written to exist inside the Operating System's kernel environment. Similar attempts have tried to execute above the kernel. An example is found in Linux using a system called FUSE (File System in User Space). FUSE is effective for this type of operation but constantly requires data to be passed between user space and the kernel in small packets causing extreme latency and slow performance. CSFS embeds itself with the kernel and not the user space. This allows the system to have low latency access by residing inside the kernel with other kernel system components producing reliable, low latency and high-speed operations. Its location above the Linux VFS is also a portable solution. In a Windows environment a filter driver is utilized mirroring the same capabilities and efficiencies as the kernel driver provides in Linux.

HIGH PERFORMANCE I/O

While in a data center or at a remote site, writing data with a REST interface causes additional development hurdles. REST interfaces require elaborate handshaking using SHA/MAC algorithms for security and MD5 algorithms for data integrity. Although adding deduplication, compression and encryption will in general cause extreme I/O latency, CSFS solves these issues by adding asynchronous I/O where file data is read or written into buffers from 4k to 500MB in size. The buffer size and the number of buffers are configurable at mount time with larger asynchronous buffers which allow file write speeds to exceed 500MB per second per instance over the wire. This is ideal for supporting multiple data centers, remote offices and mobile users as well as providing significant performance advantages for object-based storage. Asynchronous buffers are also ideal for public cloud storage that may not require fast access, but where security and deduplication are important.

DYNAMIC REAL TIME SNAPSHOT, VERSION CONTROL AND GLOBAL MULTI-USER ACCESSIBILITY


One of the most important features missing from most existing file systems is the ability to save deleted files and/or versions of files. While systems such as Windows and Mac provide a locally available option on a user by user basis, CSFS has added this capability as a standard feature of the file system. This supports natively and transparently global collaboration, dynamic global file locking and multi-user real time simultaneous modify and save. CSFS also maintains personal versions as well as providing a global view and management model to system administrators and corporate security groups. Any file that is deleted or modified is historically retained based on definable policies, allowing users and administrators real point-in-time recovery and fall back without having to recover files from their backup environment. Version policies exist, allowing administrators to state how many versions they would like to save. As additional versions of a file or document are saved, more storage is utilized. CSFS provides a multi-tier policy model that allow users to trim the version count based on a variety of parameters. The CSFS policy structure has been designed to support a user defined approach allowing for a series of multi-point policies to exist by group, user, size, file type, access date, age, etc. (An example would be unlimited point-in-time during a working day, then trim the version count to 5 versions at the end of the day. If the file was not touched after 1 month, trim to 3 versions and then trim again after 1 year to 1 or 2 versions.) This is a global capability that incorporates a multi-copy, multi-target, multi-site deduplicated disaster prevention model with administrator controls from anywhere in the world. The aforementioned capabilities completely replace all forms of traditional backup software while providing an exponential protection and safety model. The concept of recovery is replaced by the real time availability of transparent global reacquisition.

GLOBAL FILE SYSTEM

Enterprise accounts that need access to data from multiple locations will require a central file view or “single pane of glass” allowing users, based on permissions, to see and access their data in the cloud. CSFS uses a concept called Metadata Isolation where file based metadata is physically separated and maintained on different physical storage from the physical data. Metadata Isolation allows CSFS to write metadata and data in parallel to an independent instance. This can be performed locally at the user’s site, clustered across multiple user sites, or in the cloud as an intelligent management console. Additionally, all intelligence and catalog information is always saved in the object storage or other assigned storage device and cached in the Global View Management Console; the official access point for all CSFS file systems. This “belt and suspenders” approach guarantees in the event of a catastrophic failure that all catalog information and system intelligence can be rebuilt from a simple CSFS utility while still allowing high-speed access for metadata locally and remotely around the globe. A unique design characteristic of the CSFS is the ability for a user or application to directly access their data while still adhering to all permission and accessibility parameters. If the Management Console is unavailable for any reason, the system has the ability to ping each locality and re-set the full view of all new or modified data globally. This is one of the feature characteristics that directly speak to the concept of a “last line of defense” protocol.

SUB-FILE BLOCK, GLOBAL, PRE-MOVE DEDUPLICATION

Data deduplication identifies duplicate data, removing redundancies and reducing the overall capacity of data transferred and stored. Other deduplication approaches operate at the file level, while some go deeper to examine data at the block level. Determining uniqueness at either the file or block level offers benefits in the amount of reduction each produces and the time each




approach takes to determine what data is unique. CSFS uniquely combines the characteristics of both deduplication methods, creating a deduplication scheme called Sub-File Block Deduplication. CSFS mathematically breaks down files into data blocks and configures the data deduplication block size for data specific optimization when the file system is formatted. These Sub-File Blocks are then examined for redundancy versus previously stored Sub-File Blocks from other files in the system. With CSFS Sub-File Block Deduplication, data blocks are “fingerprinted” using a hashing algorithm that produces a unique, “shorthand” identifier for each data block. These unique fingerprints along with the blocks of data that produced them are indexed, compressed and optionally encrypted and then retained. Duplicate copies of data that have previously been fingerprinted are deduplicated, leaving only a single instance of each unique data block along with its corresponding fingerprint. This is accomplished globally utilizing a unique proprietary method managed by the Global View Management Console(s). Once the block fingerprint value has been calculated, the deduplication engine compares the fingerprint against all the other fingerprints that have previously been generated to see whether this block is unique (new) or has been processed previously (a duplicate). It is the speed at which these index search and update operations are performed that is at the heart of a deduplication system’s throughput. All new fingerprints along with their corresponding full data representations are transferred to the cloud in a compressed and optionally encrypted form, allowing WAN optimization in-flight and storage optimization in the cloud. CSFS allows the Hash Table to be a memory resident. The amount of memory required to hold the Hash Table is based on the amount of physical capacity being used and the deduplication block size. The Hash Table may also be swapped to disk, which does produce higher latency but will allow small memory footprints. CSFS Sub-File Block Deduplication also has special logic for optimizing virtual machine images by mathematically aligning data into a CSFS deduplication block. This special algorithm for virtual machines adds little latency to the system but guarantees all blocks are properly deduplicated. CSFS automatically invokes this optimization to maximize the deduplication of the virtual machine images.

GLOBAL CLOUD DEDUPLICATION

One of the key advantages of Cloud Storage or Geo-Spread Object Based Storage systems is the ability to access Cloud Volumes from any location around the globe. Traditionally, this has presented particular problems for data deduplication systems. How can all the deduplication data be kept in sync? HyperStore Connect for Files solved this problem by creating Global Cloud Deduplication as a feature. Global Cloud Deduplication allows all global locations to deduplicate and compress data into a common Cloud Volume. CSFS utilizes a Cloud Hash Table that resides in the Global View Management Console and is specifically assigned to the Cloud Volume. CSFS communicates with the Cloud Hash Table utilizing an RPC mechanism over a standard TCP socket interface brokering hash lookups and allocating common metadata so all file systems are capable of storing and retrieving common files in and out of a Cloud Volume. For instance, a user in New York could write a file in a deduplicated form to a bucket and a user in Chicago would be able to read it immediately after the file has been closed. The key to the underlying capability of virtual low to no global latency is the CSFS code structure managed and directed by the Global View Management Console.

DATA COMPRESSION

Data compression encodes data so that it occupies less physical storage space. Data compression algorithms search for repeatable patterns of binary 0s and 1s within data structures and replace them with patterns that are shorter in length. The more repeatable patterns that are found by



the compression algorithm, the more the data is compressed. Compression algorithms adapt dynamically to different types of data in order to optimize their effectiveness. Because data compression is based on the contents of the data stream, the algorithm is designed to adapt dynamically to different types of data. The operations performed by the algorithm that produced the compression are reversed to “decompress” compressed data. The effectiveness of any data compression method varies depending on the characteristics of the data being compressed. When combined with data deduplication, compression adds substantial savings over the wire and when data is at rest.

FILE SYSTEM CACHE

Working with data in the cloud or object storage can be expensive based on many variables resulting in time and money. Most cloud storage vendors charge little money for uploads but a substantial amount of money for downloads. Object storage generally presents large rebuild penalties as well as performance transfer penalties. In either case, the option to utilize inexpensive disk and keep a local tier saves money while maximizing performance. CSFS allows for a multiple, user defined, local and remote, mixed media tiers assignable globally or for each user defined mount point. User defined policies will recognize files based on user defined triggers and will place and manage the data based on these triggers and policies. One of the key benefits of the multi-tier, mixed media capability is to increase performance and reduce time and penalty costs associated with Puts and Gets in the public cloud and the time penalty of slower object based storage systems. One of the key challenges presented with providing a dynamic, real time global, multi-user shared architecture is whether or not the most recent version of the file is being accessed. All metadata is managed by the Global View Management Console. When a file is opened CSFS will request the last modified date, time and size of the file. If this data is different than the cached file, the local cached file will be truncated and the data will be pulled down from the global view when accessed; maintaining any version control policy that may be in place. This capability assures each user will always have access to the most current copy of a file from anywhere in the world. Mobile users also realize this capability regardless of device. The CSFS provides Administrators with simple, definable policies to manage where, how long and for what reason a file resides and how many copies of the file will exist. This global capability provides a user or IT Administrator the ability to maximize performance and accessibility while empowering them with the most granular and simple to deploy cost management controls.

AUDIT AND REPAIR, TRUE FILE IMMUTABILITY, FAILURE PROBABILITY AND PREDICTION

While most current file systems claim to have the ability to repair their internal structures, these same file systems do not have the ability to audit, repair and report on the actual stored file data. While this ability is a mandatory requirement for anyone considering to claim they have created a permanent archive or true digital library, it has been argued it is equally important for new file creation or file modify. The industry has historically proven corrupted files are backed up or migrated with no knowledge by the user. CSFS has the inherent ability to access any hashing algorithm like SHA-1, SHA-256, MD5 and others in series and in combination. The results are used to digitally fingerprint files upon first writes or write/modifies. These fingerprints themselves are then hashed, creating a cross reference identifier to provide the foundation for third party immutability – the only true approach to guaranteeing a file has not been modified. With this capability CSFS performs individual and full system audits as defined by the user. The system provides Red Flag/Green Flag reports based on these audits, and utilizing the CSFS distributed protection coding, will repair any individual file at an object level. This directly addresses the known issue of silent data corruption or

bit-rot. The architecture provides for 3rd party, independent applications to audit and compare the system fingerprints at the individual file level. CSFS allows the user to export the file and hash export table for 3rd party point-in-time immutability verification and cross system file validation. From the perspective of hardware, CSFS has the provisional capability to monitor, predict and report on the status and effective probability conditions of the hardware managed within the environment.

GLOBAL VIEW MANAGEMENT CONSOLE AND LOCALITY AWARE OPTIMIZATION

While the Global View Management Console can use CSFS locally to store and manage metadata, intelligence and catalog information for cell phones, tablets, notebooks, workstations, small businesses and personal use, the enterprise requires much more. And while this works for most file system functions, advance functions such as Tier Management, Data Movement, Globally Dispersed Data Protection, Advanced Search and Authentication require the capability to provide a globally accessible and manageable catalog/database. The concept of syncing local access points, whether in a building, across campus or around the world, is an antiquated 30 plus year approach that still today exhibits shortcoming across numerous technologies. The CSFS architecture provisions the ability to extend its current structures by adding an intelligence based catalog/database manager within the file system.

UNLIMITED PERFORMANCE SCALABILITY, IDENTITY INHERITANCE AND DIRECT ACCESSIBILITY

In the event of any physical access point or device failure, the ability to provide true seamless multipoint access transparently to users and applications is a mandatory requirement. Any single physical device or access point will always, at some point, become a performance limiting factor. CSFS provides both the ability to expand as many device access points to accommodate any user count and performance requirement while supporting the ability for any of these devices to fail or be removed – supporting a zero impact system survivability and live maintenance capability. In the event of any physical or virtual access/file system mount point becoming either intentionally or unintentionally unusable, any of the remaining access points have the ability to inherit the broadcast address of the other access point allowing for an uninterrupted, zero impact experience for users and applications. Fundamentally CSFS abstracts performance and capacity components providing true independent scalability and management. CSFS provides for truly independent front end and back parallel compute grids. Parallel path and multi-threaded CSFS provides a variety of tuning mechanisms to optimize and accelerate all aspects of a globally dispersed file and storage management system. In addition, CSFS provides the ability to present data on raw storage with natively accessible capabilities. In the area of Big Data there have been discussions over the last few years about the ability to directly access data on raw media and around the concept of open standards with the ability to support data accessibility 100 or even 1,000 years from now. The Apple iOS does not support NTFS or vice versa. Neither support the ext3 or ext4 platform. CSFS provides the capability to allow applications to directly address the question of data accessibility on raw media. Users can take advantage of the CSFS Distributed Protection algorithms, deduplication, compression and encryption with the availability of translation modules, allowing for the user's applications to interpret and access this data stored under these extended conditions. Along these same lines, CSFS provides a pass through capability to support direct REST protocol reads and technologies like OpenStack Keystone properties independent of the CSFS environment. These combined capabilities directly address and satisfy the questions by providing the answer to long term direct access data survivability.

IN FRONT OF AND BEHIND OF AUTOMATED TIER MANAGEMENT

Although users may want to adopt a new approach to data management, supporting the transition from one paradigm to another has historically proven challenging. An integral approach to the Clouidian HyperStore Connect for Files model is to provide every component necessary to empower the adoption and seamless transition to CSFS. While some users will start fresh with CSFS, many users will want to move existing data from present storage to take advantage of and fully integrate their environment. When presented with this challenge users have historically wanted to move transparently with zero impact to their existing users and applications (see parts 1-3 below).

Part 1: Clouidian HyperStore Connect for Files has developed the core technology for an outward facing data mover that provides the ability to move data from existing storage into the CSFS environment. To facilitate this transition, existing files are replaced with a system pointer, allowing the file transition to be entirely transparent. Once all of the files have been transitioned from the source, the source device can either be decommissioned or re-commissioned within the CSFS environment, re-capitalizing the user's original investment. In addition, this "in front of" capability allow users to maintain specific storage tiers outside of the CSFS environment while providing all of the inherent features such as new file creation and write/modify file identification. This allows for a Native File Format data protection capability which will replace traditional snapshots with real time, definable, dynamic snapshots, multi-device, multi-copy protection and multi-site disaster prevention; eliminating the overhead, management and cost of traditional backup. This eliminates the two step process associated with virtually all current backup and migration models in use today. In this scenario, when a file no longer needs to reside in front of CSFS, since the file has already technically been backed up inside of CSFS, the file can simply be replaced with a system pointer – requiring no additional moves and eliminating traditional migration.

Part 2: The CSFS architecture has the ability to ingest the file structure of an existing environment. Utilizing this approach CSFS is placed into the existing system architecture and interprets and assimilates the existing file structure, directories, folders, file names and system information. This process may take a few hours to a few days depending on the size of the existing system. Once complete, the user can target the CSFS mount point which now represents the original mount point; seamlessly re-pathing users and applications to the CSFS environment. This process is implemented in seconds without physical file migration while being completely transparent to users and applications.

Part 3: Leveraging CSFS, IT Administrators will now have the ability to intelligently manage their data on any type or tier of storage. Users will manage their data type, location and copy count based on policies (automated, user defined or a combination of both). This capability will not only address the traditional request of placing data on specific types of storage, this will also directly support the ability to place data within different types of protection models on the same type of storage. This approach can result in an additional 30% to 50% cost savings over any form of a traditional tiered model. Delivering this granular yet easy to manage capability will empower administrators to implement the most advanced and accurate cost conditions available.

CLOUDIAN HYPERSTORE MANAGEMENT

ONE SIMPLE WEB BASED GUI

The Clouidian Management Console (CMC) is a web-based user interface for Clouidian 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 Clouidian HyperStore system administrator, you can use the CMC to 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 perform S3 operations such as uploading and downloading S3 objects. The CMC acts as a client to the Administrative Service and the S3 Service.

CONFIGURABLE

The Clouidian HyperStore system is highly configurable. Basic system configuration is implemented by the Clouidian HyperStore installation script, based on operator input during the interactive installation process. For ongoing configuration management, the most commonly used configuration settings can be dynamically modified through the Clouidian Management Console. Beyond that, a wider range of settings can be modified by editing configuration file templates on the Puppet master node, then using Puppet to propagate your changes throughout the Clouidian HyperStore cluster, then restarting the affected service(s). You also have the option of dynamically reloading certain settings by using JMX.

PROGRAMMABLE

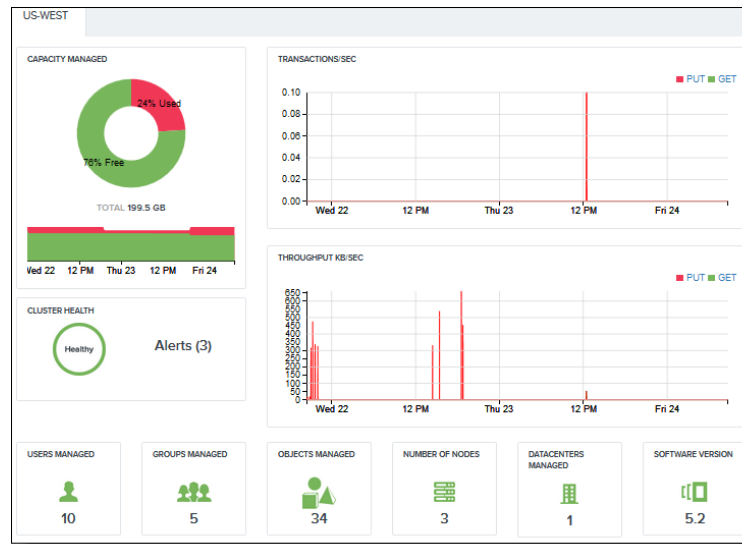
Clouidian HyperStore provides a RESTful HTTP API through which you can provision users and groups, manage rating plans and quality of service controls, and perform other administrative tasks. This Administrative API is supported by the Clouidian HyperStore Administrative Service, which runs on the same nodes as your Clouidian HyperStore S3 Service. The HTTP listening port for the Administrative API is 18081.

Clouidian HyperStore Administrative API response payloads are JSON encoded. For POST or PUT requests that require a request payload, the request payloads must be JSON encoded as well.

VIEW SUMMARY FOR DETAILS

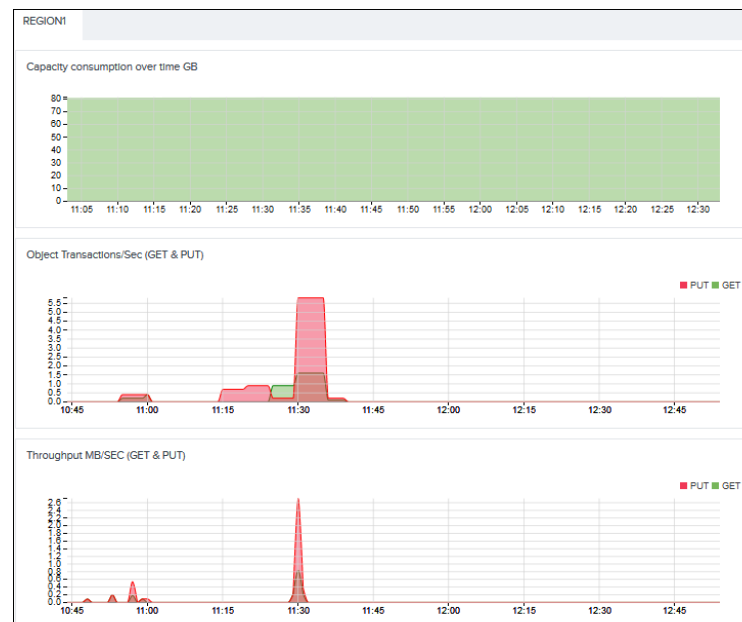
SIMPLE DASHBOARD

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.



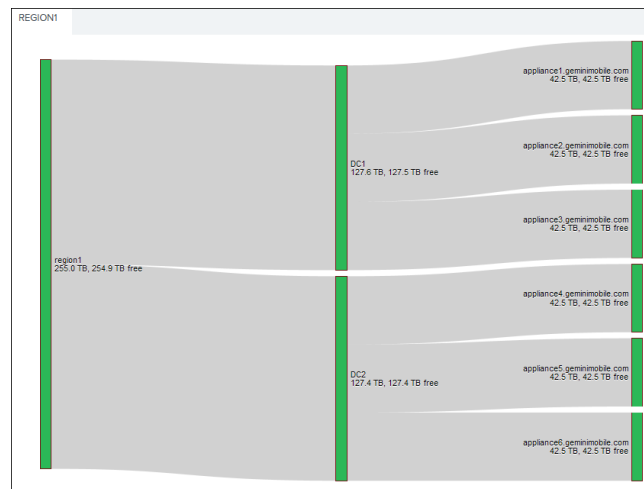
SINGLE CLUSTER USAGE & PERFORMANCE VIEW

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



CAPACITY EXPLORER

With the Capacity Explorer page you can view your available system capacity, starting with total regional cluster capacity (on the left), then moving to more granular break-downs of available capacity per datacenter and per individual node. In each case the display text indicates the total capacity designated for Cloudian HyperStore data storage and the remaining free space. For example, if for a particular node the display indicates “92.2 GB, 81.6 GB free”, then means that of 92.2GB of total disk space allocated for Cloudian HyperStore data storage on that node, 81.6GB is currently free.



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: Operation:

Report Granularity:

Group Name: User:

[List](#) [Graph](#) [Download CSV](#)

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

[Next](#) 10

STORAGE POLICIES

Storage policies are ways of protecting data so that it's durable and highly available to users. The Clouidian HyperStore system lets you pre-configure one or more storage policies. Users when they create a new storage bucket can then choose which pre-configured storage policy to use to protect data in that bucket. For each storage policy that you create you can choose from either of two data protection methods replication or erasure coding.

CLOUDIAN

[Analytics](#)
[Buckets & Objects](#)
[Users & Groups](#)
[Cluster](#)
Alerts (35)
Admin
Help

Data Centers
Nodes
Cluster Config
Storage Policies
Notification Rules
Repair Status

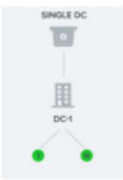
STORAGE POLICIES
+ CREATE STORAGE POLICY

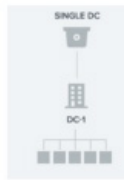
CREATE NEW POLICY

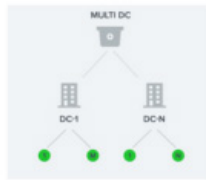
Policy Name

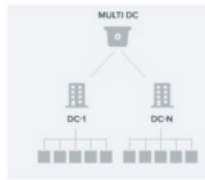
Policy Description

DATA DISTRIBUTION SCHEME

Replicas Within Single Datacenter


EC Within Single Datacenter


Replication Across Datacenters


Replication EC Across Datacenters


*Selected region does not support multiple datacenters scheme.

NUMBER OF REPLICAS

DATACENTER ASSIGNMENT

REGION	DATACENTER	REPLICA	LOCAL EC
region1	DC1	1 of 3	
		2 of 3	disable
		3 of 3	

DATA CONSISTENCY LEVEL

Strong
 Eventual
 Custom

META DATA CONSISTENCY LEVEL

Strong
 Custom

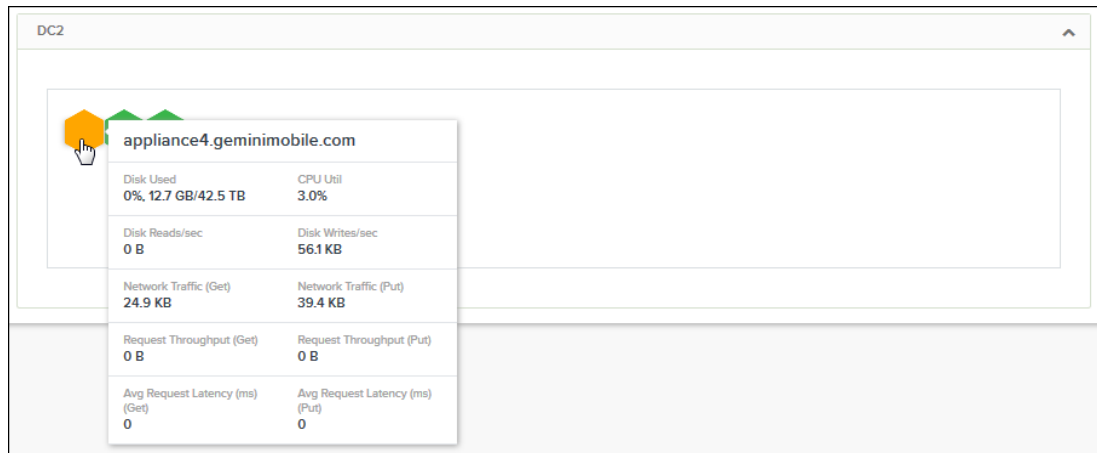
GROUP VISIBILITY

ADD

CLUSTER CONFIGURATION & MONITORING

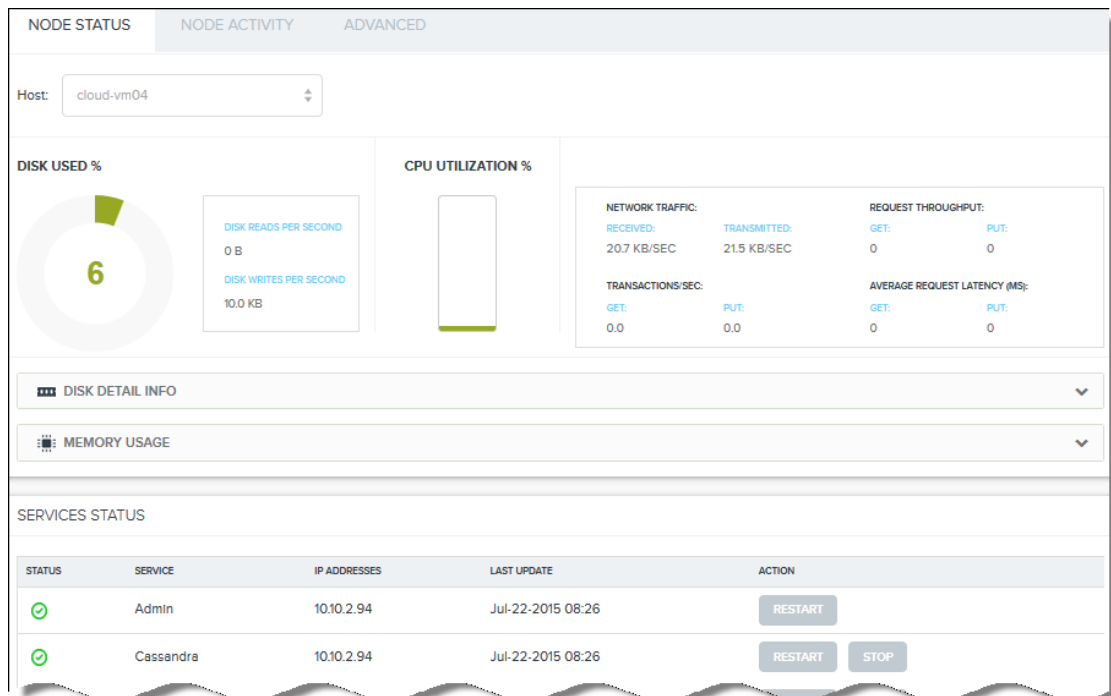
MULTI-DATACENTER & REGION VIEW

The datacenters page displays a panel for each datacenter in your Clouidian HyperStore system. For each datacenter, each Clouidian HyperStore node in the datacenter is represented by a cube.



VIEW NODE STATUS

The Datacenters page displays a panel for each datacenter in your Clouidian HyperStore system. For each datacenter, each Clouidian HyperStore node in the datacenter is represented by a cube.



CONCLUSION

Cloudian HyperStore software makes it easy to build full-featured, Amazon S3-compliant cloud storage, on-premises. It is available as either stand-alone software, or fully integrated with 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 help Cloudian HyperStore set 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 both public and private clouds, while enterprises rely on it to maintain their private and hybrid clouds.

Get started today and receive 10TB for free with our Community Edition:
<http://www.cloudian.com/free-trial/>

ABOUT CLOUDIAN

Cloudian is a Silicon Valley-based software company specializing in enterprise-grade storage. Its flagship product, Cloudian HyperStore®, is an S3-compatible storage platform that enables service providers and enterprises to build reliable, affordable and scalable hybrid cloud storage solutions. Follow us on Twitter @CloudianStorage



Cloudian, Inc. | 177 Bovet Road, Suite 450 | San Mateo, CA 94402
Tel: 1.650.227.2380 | **Email:** info@cloudian.com | www.cloudian.com

©2015 Cloudian, Inc. Cloudian, the Cloudian logo, and HyperStore are registered trademarks or trademarks of Cloudian, Inc. All other trademarks are property of their respective holders.