Qumulo Software Architecture Overview

White Paper

May 2024

Abstract

This white paper describes the fundamentals of Qumulo's software-defined Scale Anywhere™ architecture. It shows the unique approach Qumulo has taken to simplify file storage, and illustrates how Qumulo enables organizations to deploy enterprise-class file data services in a single exabyte-scale, globally consistent namespace across both on-premises and cloud-based deployments.

Table of Contents

Our goal at Qumulo is to make file storage simple for the modern, hybrid enterprise. We make it simple to secure your data. We make it simple to support demanding workflows, whether on-premises or in the cloud, at low cost. We make hybrid cloud storage simple.

Qumulo's Software Architecture

We've engineered our storage platform into a cloud-ready, scalable service that can support nearly any file-based workflow, anywhere. We also provide robust APIs to deliver automated management and real-time visibility into system and data usage. Our storage solutions meet the security and data protection requirements of Fortune 500 enterprises.

This page provides an overview of the architecture and components of Qumulo's unstructured data solution, illustrating how our product supports a wide array of use cases, from media and entertainment, to healthcare and life sciences, to cloud-based high-performance computing, to cost-effective long-term archives in the cloud.

Qumulo Architecture Fundamentals

Before diving into the individual components of Qumulo's architecture, there are several foundational assumptions that are important to enumerate:

1. Qumulo provides a 100% software-defined distributed file system that presents a single namespace. An on-premises Qumulo cluster consists of a shared-nothing aggregation of independent nodes, each node contributing to the cluster's overall capacity and performance. Individual nodes stay in constant coordination with each other. Any client can connect to any node and read and write across the entire namespace.

2. Cloud-based Qumulo instances use object storage (depending on where it's deployed either AWS S3 or Microsoft Azure Blob storage) for the data layer, in which the blocks associated with any given file are abstracted and distributed across a logical collection of discrete objects.

    a. This cloud-native architecture eliminates the legacy relationship between compute, storage and throughput, creating a fully elastic file storage service that can scale capacity to hundreds of petabytes, and can scale throughput to beyond 100 GBps.

    b. With the complete disaggregation of compute and storage that Qumulo's cloud-native architecture enables, customers have the flexibility to choose the specific levels of throughput and capacity they need, independently of one another. A customer can even deploy a Qumulo instance with an initially-low compute footprint, then temporarily scale the service's compute allocation to dramatically increase throughput for a brief period of time, then scale it back down again afterward, without at any time needing to deploy additional capacity.

3. Qumulo is optimized for scale. We ensure all aspects of our product can comfortably support petabytes to exabytes of data, billions of files, millions of operations, and thousands of users.

4. Qumulo is self-optimizing for maximum performance. Every Qumulo instance tracks data access using a heat map to identify frequently-accessed data blocks. These blocks are proactively moved by an internal prefetch algorithm: data blocks on hard disk media are moved to flash storage as their heat score increases. If the heat score continues to rise, data that is already on flash storage is proactively moved to system memory for even faster access. At a global level across all Qumulo instances for all Qumulo customers, the cache hit rate is ~95%.

5. Qumulo is highly available and immediately consistent, built to withstand component failures in the infrastructure while still providing reliable service to clients. We do this through the use of software abstraction, erasure coding, advanced networking technologies, and rigorous testing. When data is written to Qumulo's file system, the write operation is not confirmed to the service, user, or client until the data has been written to persistent storage. Thus any subsequent read request will result in a coherent view of the data (as opposed to eventually consistent models).

6. Qumulo delivers platform-agnostic file services for the public, private, and hybrid cloud. Qumulo's software makes few assumptions about the platform on which it runs. It abstracts the underlying physical or virtual hardware resources in order to take advantage of the best public and private cloud infrastructure. This enables us to leverage the rapid innovation in compute, networking, and storage technologies driven by the cloud providers and the ecosystem of component manufacturers.

7. The Qumulo management model is API-first. Every capability built by Qumulo is first developed as an API endpoint. We then present a curated set of those endpoints in

our command line interface (CLI) and the WebUI, our visual interface. This includes system creation, data management, performance and capacity analytics, authentication, and data accessibility.

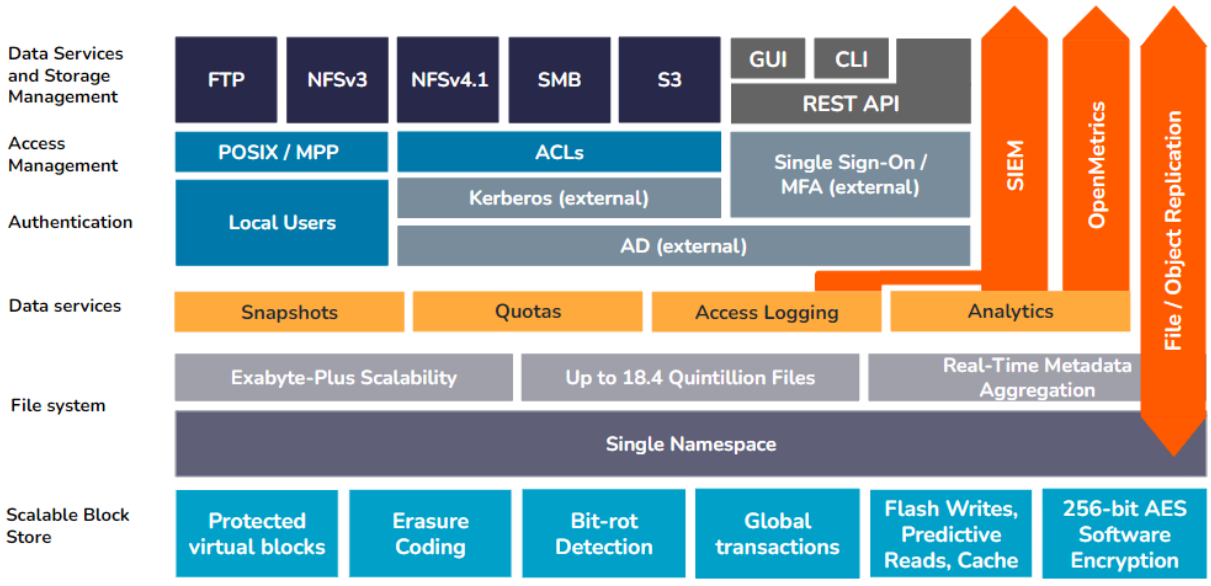
8. Qumulo ships new software rapidly and regularly. We release new versions of our software every few weeks. This enables us to rapidly respond to customer feedback, drive constant improvement in our product, and insist on production-quality code from our teams.

9. Qumulo's container-based architecture enables a unique upgrade process that minimizes disruption to users and workflows. On a rolling, node-by-node basis, the new operating software is deployed in a parallel container to the old version. Once the new instance has initialized, the old environment is gracefully shut down and the upgrade proceeds to the next node until the entire cluster has been upgraded.

10. Qumulo's customer success team is highly responsive, connected, and agile. Qumulo has the ability to connect to remote monitoring via our Mission Qontrol cloud-based monitoring service. Our customer success team uses that data to help customers through incidents, provide insight into product usage, and to alert customers when their systems are experiencing component failures. This combination of intelligent support and rapid product innovation powers an industry-leading NPS score of 80+.

Architectural Overview

Qumulo's modular architecture can be abstracted into a series of layers, with specific service controls and features bundled into each layer. These layers work together to support the scalability, performance, security and reliability of the unstructured data on a Qumulo instance, as well as the Qumulo system itself.



Data services and storage management

As an industry standard file storage service, Qumulo supports all unstructured data-access protocols: SMB, NFS, and NFSv4.1. Support is also included for object access using the S3 protocol standard, along with FTP and REST access to select data types.

System management

Any Qumulo instance, whether on-premises or in the cloud, can be managed using the same standard tools: a built-in web user interface for interactive storage and data management, a CLI-based command library, or an API-based set of management tools.

Web user interface

The Qumulo visual interface offers a web-based portal for managing a Qumulo system. The visual interface is a web-based interface, served from the system, with no separate VM or service needed. The visual interface is organized around six top-level navigation sections: Dashboard, Analytics, Sharing, Cluster, API & Tools, and Support.

Command Line Interface (CLI)

The Qumulo CLI supports most (but not all) of the API library and is focused on system administration. The CLI offers a scriptable interaction method for working with a Qumulo instance. A full list of commands can be found in our Knowledge Base (care.qumulo.com).

REST API

The REST API is a superset of all capabilities in the Qumulo data platform. From the API, administrators can:

- Create a namespace
- Configure all aspects of a system (from security such as identity services or management roles, to data management such as quotas, to data protection such as snapshot policies or data replication, to adding new capacity)
- Gather information about the target Qumulo system (including capacity utilization and performance hotspots)
- Access data (including read and write operations)

The API is “self-documenting,” making it easy for developers and administrators to explore each endpoint (and see example outputs). Qumulo maintains a collection of sample uses of our API on Github (https://qumulo.github.io/).

For more information on using the Qumulo API library, the CLI, and the web management portal, visit the Qumulo Documentation portal (https://docs.qumulo.com).

Qumulo Nexus

As Qumulo customers increasingly move to multi-cloud, multi-site enterprise operations, they need to reduce the complexity of monitoring each Qumulo instance's availability and service metrics through separate management interfaces. With Qumulo Nexus, customers can consolidate monitoring operations for all their Qumulo instances, whether on-prem, in at the edge, or in the cloud, into one management portal that delivers the same real-time analytics and data visibility as the local web interface, but through a single pane of glass.

Access management

Qumulo's software incorporates a number of inherent features and configurable controls, all designed to protect the data on the cluster.

Qumulo data security features

Every Qumulo instance, whether on-prem or in the cloud, leverages a pair of controls that ensure that all data within the file system is secured against corruption, loss, or intrusion at the data storage level.

Active Directory integration

Qumulo' security access model was engineered to leverage Microsoft Active Directory (AD) for both administrative and user rights and permissions. Besides the obvious benefits of having a single source of record for all user accounts, the use of AD for both privilege and permissions management supports industry best practices for the following:

- Seamless integration with Kerberos-based authentication and identity management protocols
- Integration with SSO and MFA access providers
- The use of Access-Control List based permissions for SMB and NFSv4.1 clients to file system data

Over-the-wire data encryption

Even with the appropriate share and data-level security settings in place, some enterprises need an additional layer of data security to protect data from unauthorized access. For those environments, Qumulo also supports over-the-wire data encryption to and from supported clients.

For SMB3 shares, Qumulo supports both cluster-wide and per-share encryption when needed. NFSv4.1 exports that require enhanced security can be configured to use either krb5i packet signatures that ensure data integrity, or to use krb5p-based packet encryption to prevent interception during transit.

All object-based traffic is automatically encrypted using standard TLS / HTTPS encryption standards.

Authentication and Access Control

Access to data in the Qumulo file system, as well as access to the Qumulo storage system, use industry-standard authentication and access protocols, ensuring enterprise-grade access management, identity control, and auditability.

Administrative Security

System-level rights and privileges are granted based on membership in one or more local groups on the individual Qumulo instance. Administrative rights are granted to all local and domain accounts that are members of the cluster's built-in Administrators group.

Domain-level administrative users

Most enterprise security policies require that the administration and management of critical enterprise systems follow a one-user, one-account policy to ensure accurate records of system access and privilege use. The simplest method for complying with this policy is by adding the relevant Active Directory user accounts to the cluster's local Administrators group.

Local administrative users

Every Qumulo instance comes with a default account, called admin, which is automatically assigned membership in the local Administrators group, and as such has full administrative rights and privileges to the cluster.

Single sign-on with multi-factor authentication

Single sign-on (SSO) eliminates the need for an administrator to re-enter their login credentials to gain access to the system. Enterprises want SSO not just because it streamlines the login process, making it more convenient for admins to authenticate, but also because it reduces the risk of account theft via keystroke loggers or interception as the login attempt traverses the network.

Multiple-factor authentication (MFA) adds another layer of security to the login process, requiring that admin users retrieve a one-time code from either a key token or a challenge request on a separate device, neither of which would be in the possession of an intruder.

Qumulo's SSO solution integrates with Active Directory via Security Assertion Markup Language (SAML) 2.0. For MFA, customers can leverage any Identity Provider (IdP) that integrates with the AD domain registered on the cluster, including but not limited to OneLogin, Okta, Duo, and Azure AD.

Access tokens

To simplify the process of automated storage and data management via Qumulo's API functionality, Qumulo offers administrators the option of generating a long-lived API token that can be used by automated workflows indefinitely, until the key is either revoked or deleted. The token is generated by an administrator via CLI, and can be attached to each API-based workflow, which can now make authenticated API calls without having to log in. For auditing purposes, each token maps to a specific AD or cluster account. If the associated user account is deleted or deactivated, the access token will stop functioning.

Role-based access control

Role-based access control (RBAC) allows administrators to assign fine-grained privileges to non-administrative users or groups who require elevated rights to the cluster for specific management tasks. The use of the RBAC model allows the secure delegation of privileges on an as-needed basis without needing to confer full administrative rights. It also enables enterprises to grant necessary system privileges while ensuring a verifiable audit trail of access and privilege use.

Data access management

Qumulo uses the same security model for managing access to file system data, using enterprise standard practices, protocols, and tools to manage and track access to all files and directories on the system

Access Control Lists

For workloads accessed via SMB and NFSv4, Qumulo supports authentication via Active Directory and Windows-style Access Control Lists (ACLs) that can be shared across both protocols.

Kerberos enhancements

All SMB and NFSv4.1 data requests, if originating from a Windows or Linux client that is joined to the same domain as the Qumulo cluster (or joined to a trusted domain), are authenticated using Kerberos-based user identity management.

Multi-protocol permissions support

Qumulo supports making the same data on the file system available over multiple protocols simultaneously. In many cases, an SMB share on the cluster may also be configured as an NFSv3 export, an NFSv4.1 export, and an object storage container. While this maximizes the cluster's flexibility, there are some considerations that need to be taken into account when it comes to managing permissions.

SMB and NFSv4.1 both use the same ACL-based permissions model, in which access is granted or denied to the user by virtue of the user's Active Directory account's membership in one or more groups whose access has been configured at the data level.

For mixed SMB/NFSv3 workloads, however, there can be a mismatch between the ACL permissions to a file or directory, and its POSIX settings. A Qumulo instance can be configured for mixed-mode operations, in which SMB and POSIX permissions are maintained separately for files and directories that are shared across both protocols.

For mixed-protocol workloads, Qumulo's proprietary multi-protocol permissions (MPP) model preserves SMB ACLs and inheritance even if the NFS permissions are modified.

Object access permissions

If a directory on the cluster is shared via S3 protocol, the directory is treated as an S3 bucket, and all subdirectories and files within that directory are treated as objects within the bucket.

When a user or workflow attempts to access an object, the system uses the access key provided by the client to identify the key's mapped Active Directory or local user ID, and then checks that ID against the object's SMB / NFSv4.1 access control list.

Management traffic restrictions

In addition to the use of SSO and MFA-based authentication of designated administrative accounts, Qumulo also supports security policies that require the restriction of admin-level access to specifically designated networks or VLANs, by offering the ability to block specific TCP ports at an individual VLAN level.

In this manner, a Qumulo instance can be configured to segment management traffic – e.g. API, SSH, web UI, and replication traffic – from client traffic, e.g. SMB, NFS and object access.

Data Services

The Data Services layer includes five management features: snapshots, replication, quotas, access logging and auditing, as well as system and data analytics.

Snapshots

Snapshots on a Qumulo cluster can be used in several ways to protect the cluster's data:

- They can be used locally for quick and efficient data protection and recovery.
- A snapshot of the live data on one Qumulo cluster can be replicated to a secondary Qumulo instance, such as an Azure Native Qumulo Cold service instance, which could support an immediate failover of file data services in the event of a systemic outage in the primary location.

- Qumulo snapshots can also be paired with third-party backup software to provide effective long-term protection (with more robust version control for changed files) against data loss.

A snapshot can be taken at any point in time, either according to a fixed schedule, or on-demand as needed. Once taken a snapshot consumes no space initially. A snapshot preserves everything in the file system – file data, directory entries, creation and modification times, permissions, etc. As files within the snapshot change over time, new data is written alongside the original version, and new entries are written in the file system identifying each version of the same file.

Snapshot locking

To provide added protection against ransomware attacks or premature deletion of critical snapshots via a compromised administrator account, snapshots can be cryptographically "locked", preventing the alteration or premature deletion of a snapshot even by an administrative user.

The use of locked snapshots requires an asymmetric cryptographic key pair, with the public key installed directly on the Qumulo instance and the private key stored externally according to the organization's own established key-management practices.

Quotas

Quotas enable users to control the growth of any subset of a Qumulo namespace. Quotas act as independent limits on the size of any directory, preventing data growth when the capacity limit is reached. Unlike with other platforms and services, Qumulo quotas take effect instantaneously, enabling administrators to identify rogue workloads via our real-time capacity analytics and instantly stop runaway capacity usage. Quotas even follow the portion of the namespace they cover when directories are moved or renamed.

Access logging and auditing

Audit logging provides a mechanism for tracking Qumulo file-system events as well as management operations. As connected clients issue requests to the cluster, event-log messages are generated describing each attempted operation. These log messages are then sent over the network to a designated remote syslog instance, e.g. an industry-standard Security Information and Event Management (SIEM) platform such as Splunk.

Real-time intrusion and ransomware detection

Qumulo has partnered with third-party providers Superna and Varonis to enable real-time monitoring of event and access logs to identify and respond to cyberattacks. To learn more about Varonis with our Azure Native Qumulo solution, visit our Varonis Integration with ANQ page. Information on Superna Ransomware Defender is available here.

System and data analytics

Qumulo's software stack is engineered to offer real-time insight into system and service metrics, including capacity and performance, in every Qumulo instance. This enables customers to troubleshoot applications, manage capacity consumption, and plan expansion (or archive) strategies. Qumulo's analytics are powered by aggregating metadata changes across the file system as they happen.

The web interface includes real-time monitoring tools for tracking system performance, capacity usage, and current activity on the local Qumulo instance. For enterprises who wish to export this information to an external monitoring solution, Qumulo supports the OpenMetrics API standard for exporting and compiling syslog data.

Replication

Qumulo's built-in replication service can copy data at scale between any two Qumulo storage instances. Besides protecting data against cyber attack, a secondary location with another Qumulo cluster can also serve as failover storage in the event of a site-level outage.

Since all Qumulo instances support the same replication features and deliver the same services regardless of location, replication can be configured to run in any direction between any two Qumulo endpoints, whether on-premises, in AWS, or on Azure.

Continuous replication

This form of replication simply takes a snapshot of the data on the source Qumulo cluster and copies it to a directory on a target cluster. As long as the replication relationship is active, the system scans any modified files to identify and copy only the specific changes to the target, overwriting any previous versions of the data.

Snapshot-based replication

With snapshot-based replication, snapshots are also taken of the target directory on the secondary cluster. Once a replication job has been completed, a new snapshot of the target directory is created, ensuring data consistency across both clusters, as well as maintaining a change log and version history for each file on the target.

The Qumulo file system

All unstructured data stored on a Qumulo file system is organized into a single namespace. This namespace is POSIX-compliant, and also supports the Access Control List standard used by the NFSv4.1 and SMB protocols, very much like other NAS systems and architectures.

Where Qumulo differentiates itself is in its ability to scale its single namespace to virtually any size, the way in which system and data analytics are inherently integrated into file-system

operations, its support for S3 as well as NFS and SMB, and its unique approach to multi-protocol permissions management.

File-system operations

Qumulo's file system was engineered from the beginning to seamlessly scale to exabyte-plus capacity in a single namespace that can host trillions of files that can be shared via standard NFS and SMB protocols. Additionally, the file system was architected with the ability to efficiently monitor file-system updates and actions, and to aggregate metadata-based statistics and operations, enabling real-time system and data analytics without resorting to resource-intensive, time-consuming tree walks.

File-system scalability

A single Qumulo instance can scale to exabytes of capacity and 2^{64} (~18.4 quintillion) files without any of the problems common to other platforms like inode depletion, performance slowdowns, and long recovery times after component failures.

Metadata aggregation

In the Qumulo file data platform, metadata such as bytes used and file counts are aggregated as files, and directories are created or modified. This means that the information is available for timely processing without expensive file data platform tree walks. The real-time analytics engine maintains up-to-date metadata summaries across the file-system namespace, collecting and updating information as changes occur. Different metadata fields are summarized to create a virtual index. As changes occur, new aggregated metadata is gathered and propagated up from the individual files to the root of the file system. Every file and directory operation is accounted for, and the resulting changes are immediately merged into the system's analytics.

Qumulo Global Namespace

The Global Namespace service offers the ability to extend Qumulo's single namespace across multiple instances, whether on-premises or in the cloud, by defining virtual data paths, called "portals," independent of the data's actual location. The use of portals means not only that data can be located on any of the customer's Qumulo deployments, it also means that users and workflows can see remote data as part of the namespace on their local Qumulo storage, and that data can be physically from one Qumulo instance to another – e.g. for follow-the-sun workflows, or to migrate cold data to a centralized Azure Native Qumulo Cold archive tier – without having to remap clients to the new path or breaking existing applications.

The first time a remote file is accessed via a GNS portal, the local Qumulo instance automatically caches a copy of the file locally. Any subsequent access of the file, whether from the same client or others in the same site, is provided from the local cache. Besides

simplifying data access across multiple on-prem and cloud Qumulo deployments, GNS also enables low-latency access to cached remote data.

The Scalable Block Store

Beneath the Qumulo file system is a protected, modular layer that serves as the interface between potentially billions (or more) of files and directories, and the physical data medium on which they're stored. In the Qumulo modular architecture, this role is filled by the Scalable Block Store layer.

Global transaction system

Since Qumulo uses a distributed, shared-nothing architecture that makes immediate consistency guarantees, every node in the service needs to have a globally consistent view of all data at all times. The Scalable Block Store leverages a global transactional approach to ensure that, when a write operation involves more than one block, the operation will either write all the relevant blocks, or none of them. For optimum performance, the system maximizes parallelism and distributed computing while also maintaining transactional consistency of I/O operations.

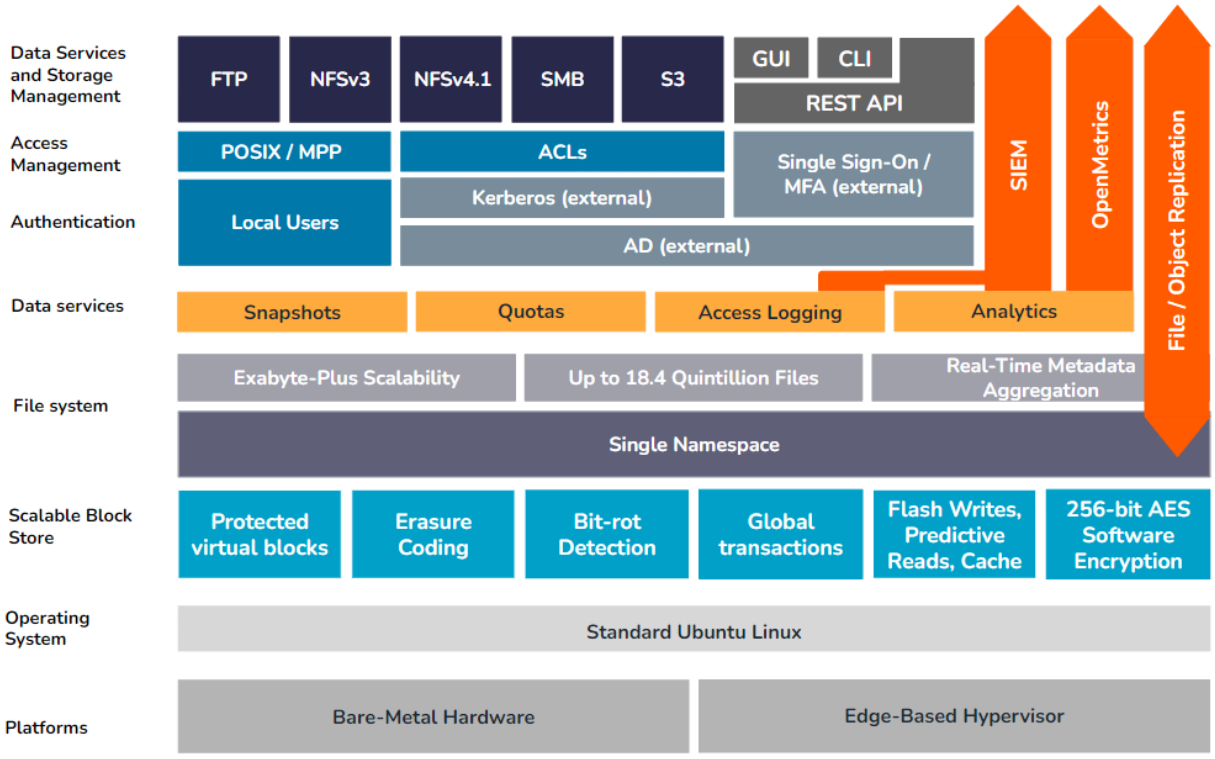
The advantage of this approach is that the absolute minimum amount of locking is used for transactional I/O operations, enabling Qumulo deployments to scale to many hundreds of nodes.

Intelligent caching and prefetching

A single Qumulo instance can store trillions of files and petabytes of capacity. However, since only a small percentage of that data is active at any given time, Qumulo has engineered several features and operations to optimize both read and write performance for active data:

1. All metadata, which is the most often read in any data set, resides permanently on the storage instance's flash tier.
2. Virtual blocks which are read frequently (as measured by a proprietary "heat index") are stored on flash, while virtual blocks which are read infrequently are moved to colder media, i.e, the system's HDD tier (if available).
3. As data is read, the Qumulo instance monitors client behavior and intelligently prefetches new data into system memory on the node closest to the client in order to speed up access times.

Physical Qumulo deployments



On a physical Qumulo cluster, the Scalable Block Store serves as the interface between the file system and the underlying storage media, which can be either solid-state flash devices (SSDs) or hard disk drives (HDDs). This layer is primarily responsible for guaranteeing data consistency across all nodes in a physical cluster, ensuring optimal performance for both read and write requests, and for providing data security, integrity, and resiliency against component failure.

Protected virtual blocks

The storage capacity of a physical Qumulo cluster is conceptually organized into a protected virtual address space. Each address within that space stores either a 4K block of data, or a 4K erasure coding hash that can be used to rebuild any data blocks lost to hardware failure. The ratio of data blocks to erasure-coding blocks is determined by the size of the physical cluster – as more nodes are added, the ratio adjusts to provide greater overall efficiency while still protecting against both disk and node failure.

In addition to the protection offered by erasure coding, the virtual block system also includes a bit-rot detection algorithm to protect against on-disk data corruption.

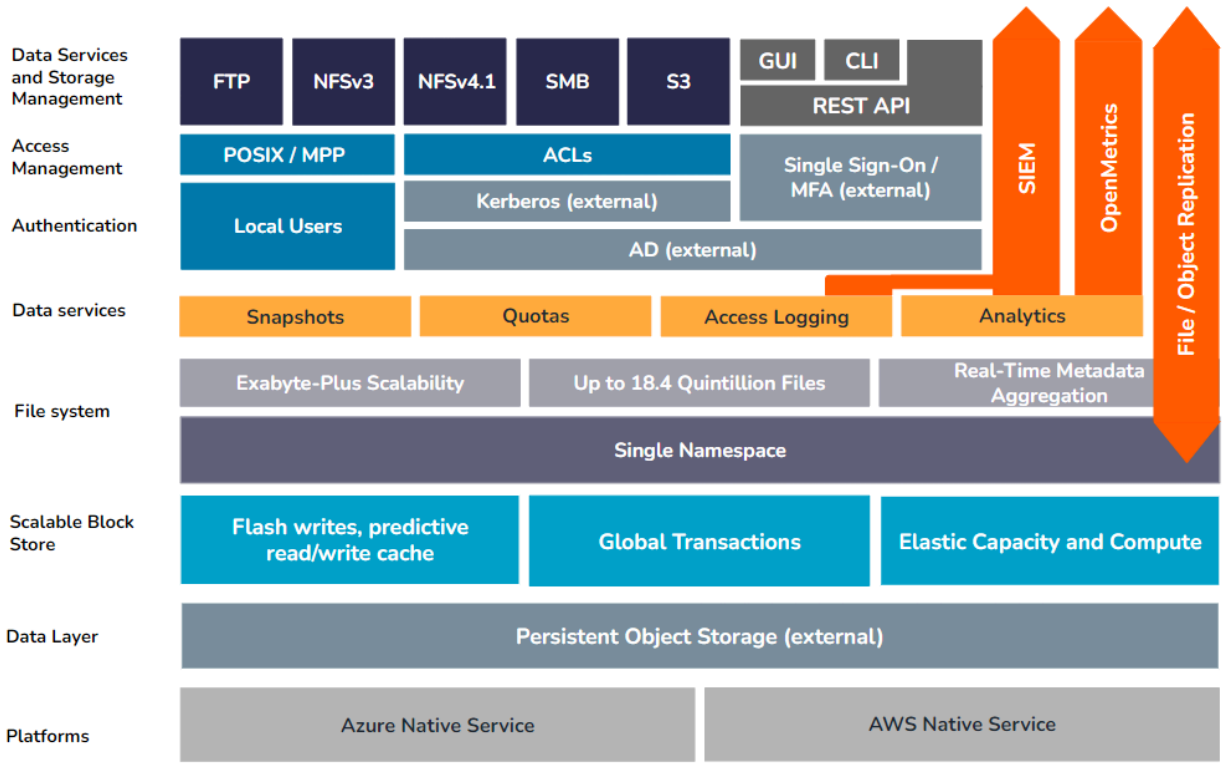
Software-based encryption at rest

On physical Qumulo clusters, the Scalable Block Store includes an AES 256-bit software-based algorithm that encrypts all file system data before writing it to the data layer. This algorithm initializes as part of the initial cluster build process, and compasses all file system data and metadata at the block level for the entire lifespan of the cluster.

Qumulo clusters in the cloud rely on block-level encryption within the cloud-storage layer, implemented and maintained by the cloud service provider and ensuring that all at-rest data on any Qumulo cloud-based instance is fully encrypted.

For enterprises that require it, the Qumulo on-prem encryption algorithm, and the encryption services provided by both Azure and AWS, support FIPS 140-2 compliance.

The Scalable Block Store on cloud-based storage



For Qumulo instances deployed on Azure, many of the functions provided on-premises by the Scalable Block Storage layer, such as on-disk encryption, erasure coding, bit-rot detection, and block management, are provided as core features of the underlying Azure Blob Storage service.

Server hardware

Qumulo's software runs on virtually any standard, enterprise-grade x86-64 based hardware, although customers looking for optimal availability and performance should consult with Qumulo directly on choosing the appropriate hardware configuration.

The underlying Linux operating system is locked down, allowing only the operations needed to perform the required supporting tasks of the Qumulo software environment. Other standard Linux services have been disabled in order to further reduce the risk surface for an attack.

Fully native software stack

Although Linux includes open-source components to provide both NFS and SMB client and server services (e.g. Samba, Ganesha, etc.), these services are not included in the hardened Ubuntu image that supports the Qumulo software environment. Qumulo develops and controls all code used for data-access protocols NFS, SMB, FTP, and S3 – in the Qumulo operating environment.

Instant upgrades

Qumulo's iterative development process is simple and streamlined, with new software updates released regularly. Not only does this enable rapid innovation to develop and roll out new features, but it also fosters a more secure storage platform.

Qumulo engineered the upgrade process to be quick and easy. Our entire software stack is containerized which enables us to upgrade an entire cluster, regardless of size, in 20 seconds, eliminating roll-backs in the process, since the functionality and stability of the updated version can be fully validated before the older version is shut down.