Qumulo Distributed, Scale-Out File System On AWS

Reference Architecture

February, 2022

Version: 1.1

Stefan Radtke,
Field CTO, EMEA
Qumulo
AWS Solution Architect Professional

Table of Contents

Introduction

The needs for Enterprise File Services in the Cloud

Companies are moving more and more applications to the cloud. One of the fastest migration methods is “lift and shift”, which means you move existing applications without major redesigns to your VPC. And because the majority of on-premises applications work with filesystems for Unix/Linux and/or Windows, Enterprise filesystems are among the fastest growing services in the Cloud. CIO’s and system administrators require that the following challenges are being addressed for the migration:

- Access to the data should be possible from any protocol at the same time
- Permissions and ACLs should be ‘translated’ transparently between POSIX and Windows and potentially other protocols such as FTP or HTTP
- The solution should have Enterprise features that Storage Administrators use with on-premise solutions, such as Snapshots, Quotas, Kerberos integration, UID/SID mapping
- At the same time, the solution should be software defined with Cloud native integration. This means, for example, automated implementation through Cloud Formation Templates or Terraform as well as integration with AWS’s CloudWatch.
- The solution should be scalable and allow expansion of capacity and performance in real time without any service interruption.
- The system should be able to deal with billions of files without the requirement of performing treewalks for certain operations such as backups, analytics or the creation of usability statistics.
- They want a single solution for SMB, NFS and sometimes FTP
- Some companies have a multi-cloud strategy. In this case they wish to have a similar file solution with the same APIs, management, cloud integration, performance tiers, backup methods, access protocols etc.
- Ideally, the solution allows moving data between the filesystem and AWS Simple Storage Service (S3) because in many cases, their central data repository lives in S3. Or they have data on the filesystem that they want to process with AWS Service that operates on data in S3.
- Workloads that rely on on-premises filesystems can be a challenge to move to the cloud without a filesystem that works in a hybrid environment.

- Ideally, the solution would provide real time performance and capacity analytics to gain insight on usage patterns, utilization and cost optimization.

Qumulo as a Cloud Filesystem

Qumulo Cloud Q is a cloud-native solution that is built on top of Amazon Elastic Compute Cloud (EC2), Amazon Elastic Block Store (EBS) volumes and Amazon Simple Storage Service (S3). It delivers features that addresses the aforementioned needs:

- It's a single solution for your VPCs, your Regional Zones and AWS Outpost
- Scale-out architecture: scales up to 100 instances, currently about 30+ PB in a single namespace
- Ultra high aggregated throughput with low latencies at around 1ms on average
- Flexible configurations to meet the needs of High Performance workloads or low performance active archives
- Multi-protocol: files can be accessed through NFS/SMB/FTP/HTTP simultaneously
- Native and directory based copying of file data into an S3 bucket and reverse
- Fully-programmable API
- Advanced CFT for automated deployments provided
- Kerberos/Active Directory and LDAP integration
- Snapshot integration
- Real-time quotas
- Multi-cloud replication and on-premises to AWS replication

The Infrastructure in AWS

Deployment Overview

The Qumulo Core file system is built as a user space application that runs on top of a stripped down Ubuntu LTS version for which Qumulo provides updates on a bi-weekly cadence. It is a clustered system starting from 4 nodes and scaling to 100 nodes to date. The smallest cluster can be as small as 1 TB while the largest deployment can currently host 30.5 PB of data.

The following picture illustrates a minimal stack that is deployed through the Cloud Formation Templates launched by the AWS Quickstart for Qumulo Cloud Q [7]. This reference deployment meets the AWS Well Architected Framework principles.

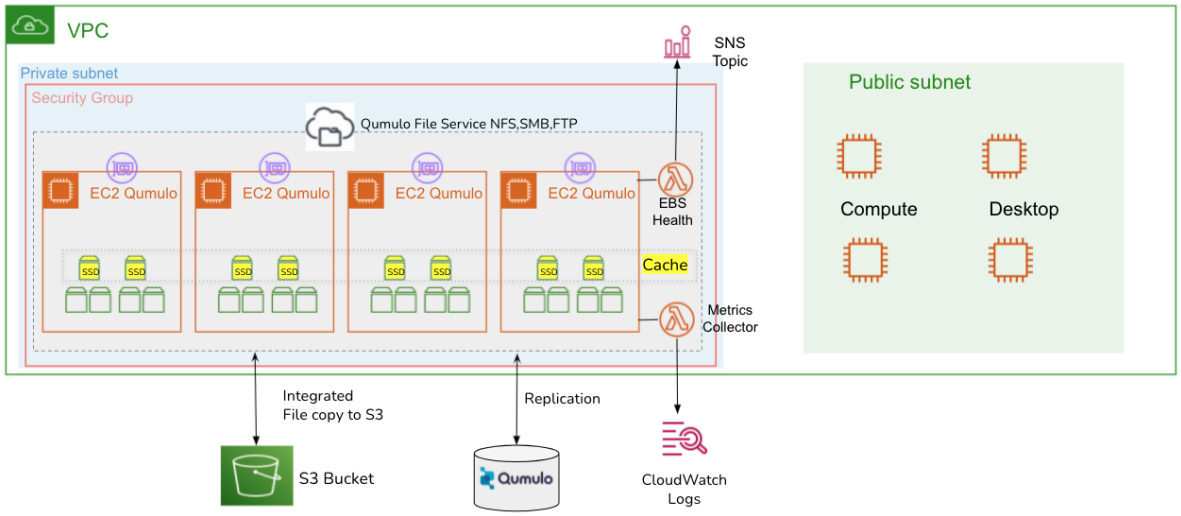

Figure 1: Minimal Qumulo cluster deployed in a private subnet

Currently, supported instance types are m5 and c5n. The instance type determines the performance to a large extent (more about performance later). The storage space is made up of EBS volumes. Depending on the node type, volumes are either GP2 volumes (all flash nodes) or a mix of GP2 and SC1 or ST1 (hybrid nodes). Each node gets a static internal IP address and typically 3 floating IP addresses that fail over to the remaining nodes if one node should fail. Optionally, the cluster could also be configured with an Elastic IP per node if public IP-addresses are needed.

A Lambda function will be deployed to health check all EBS volumes and automatic replacement if one or more EBS volumes fail. Another Lambda function gathers detailed metadata metrics from the cluster and stores them in CloudWatch logs.

Deployment Automation

There are currently three options to deploy a Qumulo cluster in AWS in an automated way:

1. By using the AWS Quickstart for Qumulo Cloud Q. It is an automated reference deployment built by Amazon Web Services (AWS) and Qumulo. The underlying AWS CloudFormation Templates automate all required steps to build a Qumulo Cluster according to best practices so that you can build and start using your environment within minutes.
2. The CloudFormation Template that is provided of each Cluster type in the AWS Marketplace.
3. The Terraform Templates provided by Qumulo on Github: https://github.com/Qumulo/aws-terraform-cloud-q

The following table lists the different features that are being deployed. As all of them are being constantly developed, please check for actual functionality before you deploy.

What's deployed by the Stack	AWS Marketplace CFT	AWS Quickstart for CloudQ	Terraform
Basic Qumulo Cluster	x	x	x
Sidecar Lambda Functions	x[1]	x	x
Cluster IAM Role		x	x
Cluster Placement Group		x	x
Cluster Node Tags		x	x
EBS Volume Tags		x	x
Cloud Watch Resource Group		x	x
Cluster Floating IPs		x	x
Cluster Sidecar Config		x	x
Local Zone Awareness		x	x
Deploy on Outpost		x	x

[1] The CFTs for the sidecar function needed to be launched separately

KMS CMK Policy Mod for Sidecar		x	x
Auto Software Upgrade at Creation		x	x
Automated Node Adds		x	x
Admin Password in Secrets Manager		x	x
Route53 Private Hosted Zone DNS		optional	optional
Public Mgmt IP & Replication		optional	optional

Table 1: Automated Deployment Options

Please note:
The description of the reference architecture discussed in this document assumes that the Qumulo deployment uses the first option listed above: The AWS Quickstart for Qumulo Cloud Q

Networking

VPC and Subnets

The Qumulo Cluster will be deployed in the customer's VPC. As a best practice, a Qumulo cluster will be deployed in a private subnet. During the deployment, the VPC and subnets for the deployments can be freely chosen.

Security Groups

If the Qumulo provided CFTs are being used for deployments, the Security Groups will be automatically configured for you.

The SG rules will allow traffic to and from the cluster for the following services:

- NFS
- SMB
- FTP
- HTTPS
- SSH
- REST
- Qumulo Replication (Standard port is TCP/3712)

Client Traffic Distribution

Each node in a Qumulo cluster has two address types:

1. **A persistent IP address.** This address is fixed and stays permanently on a node. These addresses are used for management but should not be used for client traffic.

2. **A number of floating IP addresses.** If a node fails, the floating IP addresses of the nodes will be distributed to the remaining nodes in the cluster. As best practices, the number of floating IP addresses deployed per node should equal (n-1). For example, in an 8 node cluster, each node would bet 8-1=7 floating IP-addresses. This would guarantee optimal equal distribution of the IP-addresses from a failing node to all other nodes and thus, traffic from clients would still be equally distributed. However, in clusters with moderate client traffic you might be fine to opt for just 3 floating IP-addresses per node.

The Client requests accessing the filesystem should use the floating IP addresses. The requests are typically distributed using a round robin DNS configuration. Each floating IP address would get a separate A record in DNS. If Active Directory is being used for Authentication, we highly recommend configuring the Active Directory server also as your DNS server.

The article Configure DNS Round Robin on a Windows Server for Qumulo Core describes the method.

Alternatively, Route53 can be used to configure the DNS records for the cluster. If the Quickstart [7] is being used for deployments, you'll be asked which option you choose. For a more detailed discussion of the DNS options, please refer to [9].

Access for Management

The WebUI and API endpoints can be reached through HTTPS on any floating or persistent IP address of the cluster. If your management client is external to the VPC, an optional public Elastic IP address can be configured on an AWS Network Load Balancer. It is not recommended for production workloads because your cluster management is reachable from the Internet (you can protect your NLB and EC2 instances with AWS Shield Advanced against DDoD attacks though). A better solution is a VPN connection from your client network to the VPC or at least a jump server within the VPC.

Remote Monitoring

Qumulo provides a remote monitoring service that is accessible via https://trends.qumulo.com. It provides detailed visualizations of historic telemetry data from the cluster such as capacity, IO characteristics such as throughput, IO size, latency, number of clients, system temperatures, network traffic distribution, capacity forecast and much more. For this service to function, an outbound HTTPS connection on port 443 is required. The Security Group will be configured automatically by the CFT while the VPC firewall rules need to be updated by the administrator. More details on the Remote Monitoring Service are outlined in this article: Qumulo's Cloud-Based Monitoring.

Security

Qumulo Security Best Practices

Qumulo offers a variety of security features. These features can be placed in three categories: Preventive controls, detective controls and recovery features.

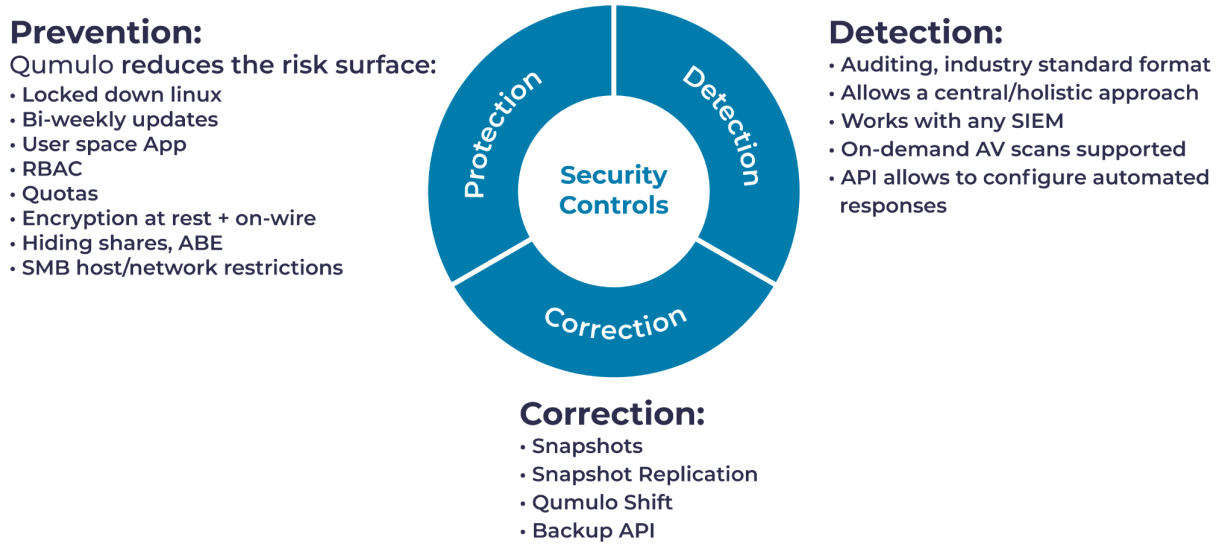

Figure 2: The three categories of Qumulo's security features

These features and the security best practices are not AWS specific and are therefore described in detail in a separate white paper: Qumulo Security Architecture and Best Practices to Counter Malware. The AWS specific security measures will be covered in the next section.

AWS specific security measures

The AWS specific security measures described in this section are automatically deployed with the Cloud Q Quickstart.

IAM Permissions

If you use the Cloud Q Quickstart, make sure you sign in to the AWS Management Console with IAM permissions for the resources that the template deploys and the services it leverages. The AdministratorAccess managed policy within IAM provides sufficient permissions, although your organization may choose to use a custom policy with more restrictions. The following AWS services are required in a custom IAM role or IAM user to deploy this template:

application-autoscaling:*	applicationinsights:*	autoscaling:*
cloudformation:*	cloudtrail:*	cloudwatch:*
compute-optimizer:*	ec2:*	elasticloadbalancing:*
events:*	health:*	iam:*
kms:*	lambda:*	logs:* resource-groups:*

AWS Key Management Systems

All data on Qumulo is encrypted at rest at the EBS layer. For the EBS volumes, you can either decide to use a key that is created and managed by AWS or you can provide your own key that is then being stored in the AWS KMS.

AWS Secrets Manager

The AWS Secrets Manager is being used to store credentials such as

- The Qumulo username/password for administrative access
- The password for Qumulo Software download access
- The Qumulo username/password for the Sidecar Lambda functions

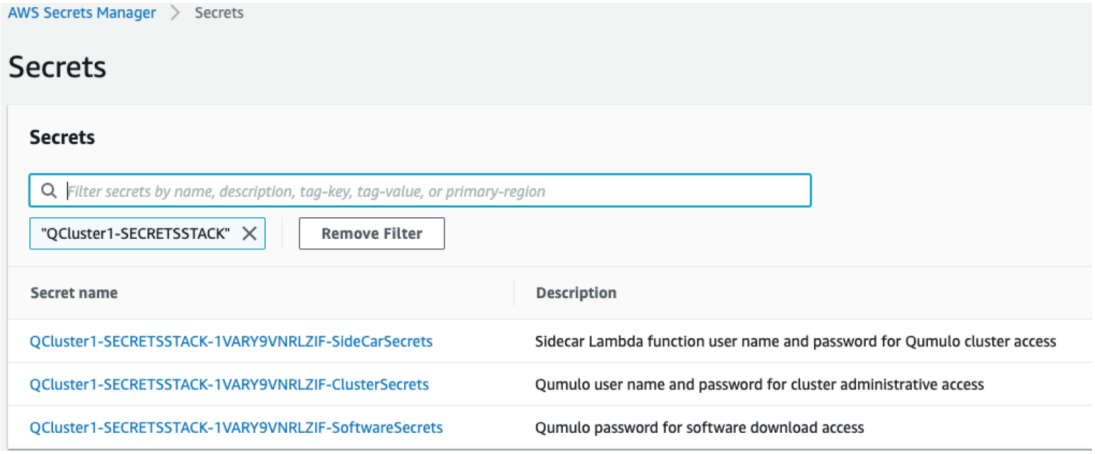

Figure 3: AWS Secrets Manager storing several Qumulo usernames/passwords

AWS Parameter Store

The Quickstart is storing some cluster parameters in the AWS Parameter Store:

- Tracks software versions,
- Cluster IPs,
- Instance IDs
- UUID

Also, the 'last-run-status' of the provisioning instance is stored here.

AWS Permissions Boundary Policy Name

During the deployment with the Quickstart for Cloud Q, you can optionally assign an IAM Permissions Boundary Policy to restrict the IAM roles created for the Qumulo cluster EC2 instances and the provisioning instance to the desired boundary. This is not necessary and is optional.

Care must be taken that the boundary policy is not overly restrictive or features and functions of Cloud Q on AWS may be impacted. The IAM roles created for these resources conform to a least privilege model.

AWS CloudWatch Logs for Qumulo Audit Messages

The Quickstart for Cloud Q offers the option to enable Qumulo Audit logs being sent to CloudWatch. Alternatively, this can be enabled/disabled at any time. For further detail refer to the article [10] Qumulo in AWS: Audit Logging with CloudWatch.

Termination Protection

It is highly recommended to enable both EC2 termination protection and CloudFormation stack termination protection. This guards against any accidental deletions of the cluster or an instance that may result in data loss.

Performance and Scalability

Single stream throughput

The single stream throughput for reads and writes is limited to 600MB/s or lower if an instance type and EBS configuration won't support that upper bound. This number equates to the AWS 5 Gbps single TCP flow rate-limit enforced outside of an EC2 placement group. This value could only be exceeded if cluster nodes and compute nodes are deployed in the same placement group (by default, the Quickstart template deploys into a cluster placement group to minimize latency between the cluster nodes).

Multi-stream throughput

The multi stream throughput varies with EBS volume configuration and EC2 instance type. Smaller instance types have less network bandwidth and less EBS bandwidth subjecting them to burst credits. Smaller EBS configurations are also subject to burst credits. For guaranteed performance, respective of baseline IOPS, choose at least a c5n.4xlarge instance type. Then adjust the instance type to increase throughput. All flash architectures should be chosen for high throughput workloads, especially in smaller usable capacity clusters, or highly random workloads. IOPS is another factor to consider for small file workloads or small usable capacity clusters.

The following graph shows the multi-stream performance for an all-flash configuration where each node hosts 8 TiB of data (please be aware that the y-axis shows the throughput in MB/s on logarithmic scale):

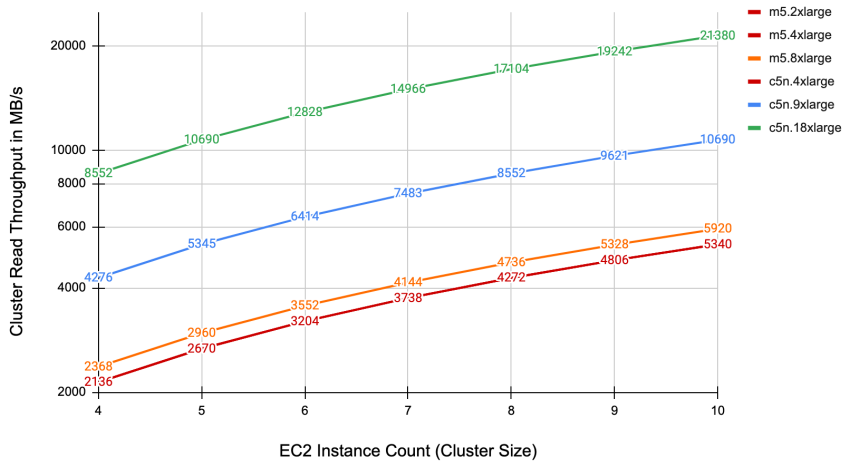

Figure 4: Qumulo Cloud Q All-Flash max read performance per cluster and node count for different instance types. Source: [1]

The following statistics shows the aggregated read latency across the Qumulo global install base. This global install base contains roughly 70% of hybrid nodes (HDD and SSDs) of clusters in the cloud and on-premises. Even with the majority of nodes hosting data on HDDs, 90% of all read requests are being served with latencies smaller than 1ms. This is a result of Qumulo's intelligent predictive caching algorithm. It enables fast reads and identifies I/O patterns and pre-fetches subsequent related data from disk into SSDs or memory.

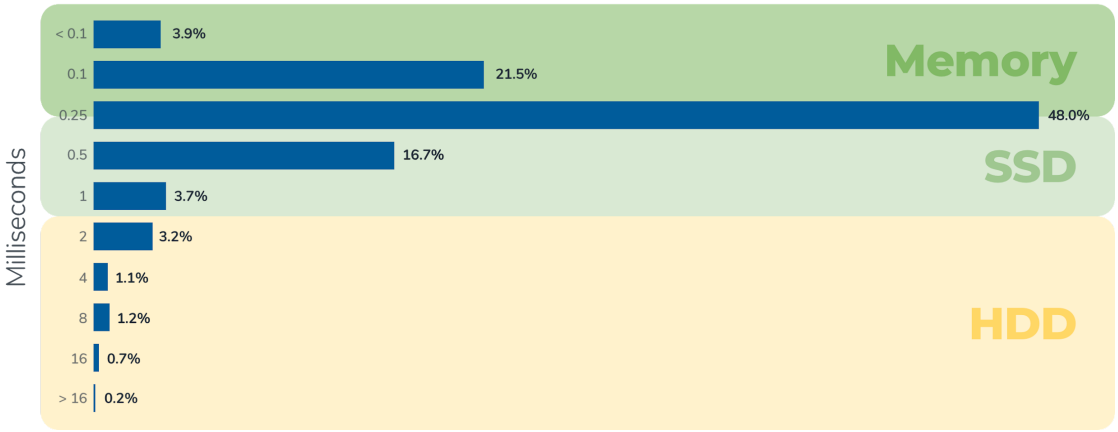

Figure 5: Aggregated read latency across Qumulo's global install base

For a detailed discussion of performance characteristics, please refer to the White Paper Qumulo Clusters in AWS, Sizing, Performance, & Cost [1].

Reliability

Qumulo's Scalable Block Store

The Qumulo file system is built on top of a powerful, state-of-the art data management system called the Qumulo Scalable Block Store (SBS). SBS uses the principles of massively scalable distributed databases, and is optimized for the specialized needs of file-based data.

The Scalable Block Store is the block layer of the Qumulo file system, making that file system simpler to implement and extremely robust. SBS also gives the file system massive scalability, optimized performance, and data protection.

SBS provides a transactional virtual layer of protected storage blocks. Instead of a system where every file must figure out its protection for itself, data protection exists beneath the file system, at the block level.

Qumulo's block-based protection, as implemented by SBS, provides outstanding performance in environments that have petabytes of data and workloads with mixed file sizes. SBS has many benefits, including:

- Fast rebuild times in case of a failed disk drive
- The ability to continue normal file operations during rebuild operations
- No performance degradation due to contention between normal file writes and rebuild writes
- Equal storage efficiency for small files and large files
- Accurate reporting of usable space
- Efficient transactions that allow Qumulo clusters to scale to many hundreds of nodes
- Built-in tiering of hot/cold data that gives flash performance at archive prices.

To understand how SBS achieves these benefits, please refer to [1].

Disk Drive and Node Failure Protection

Qumulo's distributed file system has modernized erasure coding storage methods with block-based protection which is being performed by the SBS. The beauty of it is that it divides the physical disk space into virtual address spaces called pstores (protected stores). pstores consist of 4k block addresses and erasure coding is happening at pstore level. A pstore is typically 70GB in size which means that protection and well as recovery actions take place in parallel and are very fast. The design also enables the filesystem to use 100% of the net capacity without sacrificing performance (a very typical fact with almost all filesystems in the market).

For a detailed understanding of the erasure coding implementation in the SBS, please refer to [12]

SBS secures data that resides on EBS volumes through this efficient erasure coding technique. If an EBS volume fails, the protection system will kick in to guarantee data persistence even in the degraded mode. Depending on the configuration, the cluster can handle from one to four drive failures and will still serve data if 1 or more drives fail. Once failing EBS volumes are replaced Qumulo's restriper kicks in, calculates the missing data through parity information and rebalances data to the new volumes.

The same protection scheme even protects against node failures. Data is distributed by SBS in a way that the cluster can sustain one to four node failures - depending on the cluster size and configuration. In case of a node failure (failing EC2 instance), no data needs to be rebuilt since the EBS volumes would still exist. They needed to be attached to a new EC2 instance which makes the process very fast because no rebuild or data movement needs to happen.

Lambda Function for Automatic EBS Volume replacement

In a rare case an EBS volume becomes unusable by the Qumulo filesystem, the EBS volume doesn't need to be replaced manually. If the Cloud Q Quickstart] template is being used for deployment, a Lambda function called Sidecar is being deployed automatically that monitors for EBS failures. If a volume fails, the Sidecar Lambda will automatically kick in and replace the volume. The recalculation and rebalancing of data will then be managed by SBS as usual. In addition, an SNS topic can be configured so that the admin team gets informed on the event automatically.

For further detail on the automatic replacement of EBS volumes please refer to [13].

Multi Availability Zone Protection

To protect a Cluster against a complete AWS Zone outage, Qumulo provides an efficient replication mechanism that can be configured on a per directory basis. The target cluster that is being used for Disaster Recovery could be in a different Availability Zone, different region and even a different Cloud provider or an on-premises Qumulo Cluster. In case of a disaster (the primary AZ is down), the replication relationship would be stopped, which makes the target directories writable. Client traffic can then be redirected to the secondary cluster and operation can continue. The replication can be continuous, snapshot based or both.

For a detailed instruction on Qumulo replication please refer to [14].

Multi-Protocol File Access

Qumulo Cross-Protocol Permissions (XPP) automatically manages file access permissions across protocols. XPP enables mixed SMB and NFS protocol workflows by preserving SMB ACLs, maintaining permissions inheritance, and reducing application incompatibility related to permissions settings.

XPP is designed to operate as such [2]:

- Where there is no cross-protocol interaction, Qumulo operates precisely to protocol specifications.
- When conflicts between protocols arise, XPP works to minimize the likelihood of application incompatibilities.
- Enabling XPP won't change rights on existing files on a filesystem. Changes may only happen if files are modified while the mode is enabled.

XPP maintains an internal set of ACLs for every file and directory which can contain many ACEs and thus, builds a complex rights structure, just like Windows or NFSv4. These internal ACLs are called QACLS. Once a file gets access through SMB or NFS, the permissions are being translated or enforced in real time to the appropriate protocol permissions.

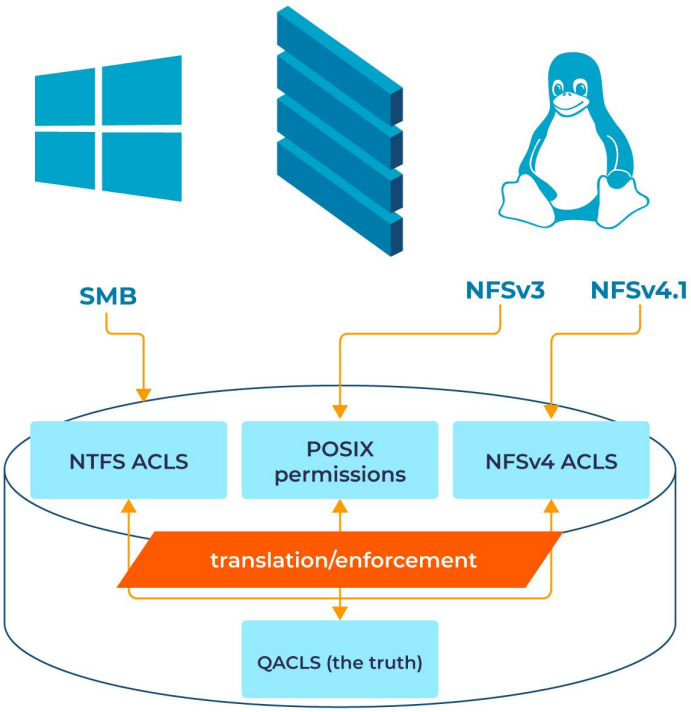

Figure 6: Translation/Enforcement for QACLS to NTFS ACLs or POSIX Permissions

Qumulo provides a set of tools that work together to query the internal QACL structure. For example, the CLI command qq fs_get_acl will provide a list of actual QACLs of a given file or directory:

```
# qq fs_get_acl --path /
Control: Present
Posix Special Permissions: None

Permissions:
Position  Trustee      Type     Flags  Rights
========  ===========  =======  =====
================================================
1         local:admin  Allowed         Delete child, Execute/Traverse, Read, Write
file
2         local:Users  Allowed         Delete child, Execute/Traverse, Read, Write
file
3         Everyone     Allowed         Delete child, Execute/Traverse, Read, Write
file
```

Another interesting command is:

```
#qq fs_acl_explain_posix_mode --path /
```

The output will explain in detail how Qumulo produced the displayed POSIX mode from a file's ACL (Please refer to [4] to study an output example).

Qumulo's multi-protocol implementation is unique and simplifies file based solutions significantly:

1. Similar deployment for SMB, NFS
2. It is very simple and just works
3. The tools provided allow a detailed understanding of the ACL and POSIX permission structure and their relations.
4. Avoids file redundancy and moving data around for different clients.

Imagine even a multi-cloud or hybrid solution without multi-protocol support. You'd have at least four different solutions to maintain with different management, different performance characteristics, different APIs, different pricing structures etc. Qumulo brings that all down to one solution for all NAS protocols, on-premises and cloud.

Moving Data Between Qumulo and Amazon S3

There is a growing number of workflows where data needs to be accessed through S3 and a Filesystem. For example, as a media content editor or artist, you typically use a shared filesystem to merge specialist effects or collaborate with other artists. Then, you might use other AWS services for transcoding files that sit in an S3 bucket. Another example is Genome Sequencing, where sequencers write to SMB, analytic processes read the data through NFS and archiving is being done on S3.

So data mobility becomes more and more important. Qumulo has a built-in feature called **Qumulo Shift** which allows data administrators to create a relationship between a directory and an S3 bucket. On demand, data can then be copied from the directory to the S3 bucket. The location of the Qumulo cluster is irrelevant in this case. It could be a cluster in your VPC, on-premises, or in another cloud.

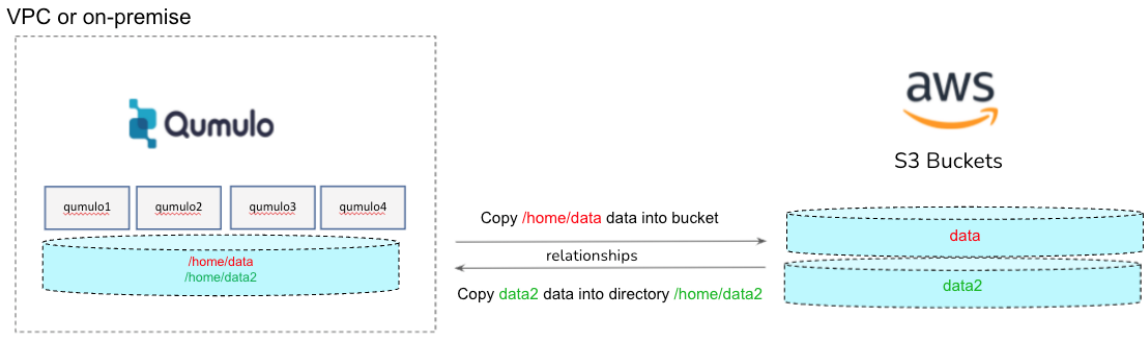

Figure 7: Copy relationships between directories and S3 buckets

To create a Qumulo Shift relationship (see a demo here), Qumulo verifies that the specified source directory exists on the file system and that the S3 bucket exists and is accessible via the specified credentials. Once the relationship is created successfully, a job is started using one of the nodes in the cluster. When performing multiple Shift operations, multiple nodes will be used. This job takes a temporary snapshot (named replication_to_bucket_<bucket_name>) of the source directory to ensure that the copy is point-in-time consistent. Shift then recursively traverses the directories and files in that snapshot, copying each file to a corresponding object in S3.

File paths in the source directory are preserved in the keys of replicated objects, i.e., the file /my-dir/my-project/file.txt will be uploaded as the object:

https://my-bucket.s3.us-west-2.amazonaws.com/my-folder/my-project/file.txt.
The data is not encoded or transformed in any way, but only data in a regular file's primary stream is replicated (alternate data streams and file system metadata such as ACLs are not included). Any hard links to a file within the replication source directory are also replicated to S3 as a full copy of the object, with identical contents and metadata—however; this copy is performed using a server-side S3 copy operation to avoid transferring the data across the internet.

When copying, Shift will check to see if a file was previously replicated to S3 using Shift. If the resulting object still exists in the target S3 bucket (and neither the file nor object have been modified since the last successful replication) its data will not be re-transferred to S3. Shift will never delete files in the target folder on S3, even if they have been removed from the source directory since the last replication.

For more information on this data mobility feature, including a step-by-step instruction to configure the relationship between a directory and bucket, please refer to [5].

A new feature, called Shift-From has been released with Qumulo Version 4.2.3. This feature allows data administrators to create relationships where the S3 bucket is the source and a Qumulo directory as the target. For details please refer to [6].

Enterprise File System Features

There are many other filesystem features Qumulo provides, such as:

- Real time analytics for capacity and performance data
- Real time quotas
- Backup integration API used by Backup and Archive solutions for fast incremental forever backups.
- OpenMetrics API for integration into 3rd party management and monitoring solutions
- Active Directory Integration

Summary

Qumulo simplifies migrations to the Cloud where unstructured data is being stored in filesystems, regardless whether data access is through SMB, NFS, FTP or HTTP. File locking and access control works across all protocols and thus, redundant data placement for each protocol can be avoided. The solution can deliver tens of GB/s throughput with latencies between 0.5-5 ms. It allows easy data movement between the filesystem and AWS S3 buckets. It integrates in a cloud native way through deployment templates and can be subscribed through the AWS marketplace.
A number of unique features that come standard with a Qumulo Cloud Q software subscription make it an attractive choice for many workflows.

We urge you to learn and be curious with AWS and Qumulo services. You can find more information here: Qumulo in AWS: Getting Started and the AWS Quickstart for Qumulo Cloud Q. Another option is to use the Qumulo Studio Q Quickstart which spins up a complete post-production environment in the cloud for remote video editing and it includes a Qumulo cluster and Adobe Creative Cloud for editing. Also, Qumulo can also be deployed as an AWS Nimble Studio option for the filesystem.

References

[1] Qumulo Cloud Q QuickStart–Sizing and Performance on AWS, by Dack Busch, Qumulo, July 2021.
[2] Cross-Protocol Permissions (XPP), KB Article
[3] Cross-Protocol Permissions (XPP) in Common Scenarios, KB Article
[4] Cross-Protocol (XPP) Explain Permissions Tools, KB Article
[5] Qumulo Shift for Amazon S3, KB Article
[6] Using Qumulo Shift-From for Amazon S3 to Copy Objects, Qumulo Documentation
[7] AWS Quickstart for Qumulo Cloud Q, AWS Quickstart
[8] Qumulo Security Architecture and Best Practices to Counter Malware, White Paper
[9] DNS options in AWS to enable IP Failover and Client Distribution, White Paper
[10] Qumulo in AWS: Audit Logging with CloudWatch, KB Article
[11] The Qumulo Scalable Block Store (SBS), Blog Post
[12] How to Implement Erasure Coding, Blog Post
[13] Qumulo in AWS: Automatic EBS Volume Replacement, KB Article
[14] Replication: Continuous Replication with 2.11.2 and above, KB Article