Qumulo on Azure Security Overview

Security Architecture

June 8, 2021

Version: 0.1

Qumulo on Azure Overview

Qumulo on Azure is a cloud-based, hosted Software-as-a-Service (SaaS) offering that provides Qumulo software and supporting components as a service (the "Service") which allows customers who prefer a "cloud first" strategy to utilize a cloud-based service, in which software and associated maintenance operations can be licensed as a comprehensive service managed by Qumulo.

With Qumulo on Azure, Qumulo provides and maintains the platform that enables customers to store files within Azure, without the overhead of setting up, hosting and maintaining the environments. The customer does not manage or control the underlying infrastructure (such as network, servers, operating systems, storage or Qumulo software components), but retains control over their files and data.

This document describes the Service as well as policies applicable to the Service, and is intended for anyone interested in researching and planning for the Service Offering. The Service is provided under Qumulo's SaaS Subscription Terms and Conditions.

Security is an integral part of our organization and our mission to provide exceptional service for our customers, and supporting this mission is central to our software development and service operation practices. This paper has been written to help you better understand the collection of security controls we have for Qumulo on Azure from both the customer's and Qumulo operation's perspective.

Qumulo Software Architecture

The Qumulo File Data Platform is a scale out, software-only, NAS (Network Attached Storage) architecture. As such, Qumulo presents standard network protocols such as the Windows Server Message Block (SMB) protocol and the Unix/Linux Network File System (NFS) protocol to clients over a standard IPv4 or IPv6 connection. Qumulo provides several data services such as snapshots, replication, quotas, auditing, and role-based access control to protect your data.

As clients connect to the Qumulo cluster, they issue requests to specific files for common read/write/modify/delete operations. The file system fulfills the requests and serves the files back to the customer over the protocol used by the client issuing the request (SMB/NFS).

Multiple “nodes” running Qumulo Core are combined to create a scale-out NAS cluster and a single volume (a single name-space). Connections are distributed between nodes to optimize performance and capacity.

Files written into Qumulo are broken into smaller blocks of data, automatically encrypted and distributed across the nodes in the cluster using a modern erasure coding algorithm.

Qumulo File Data Platform

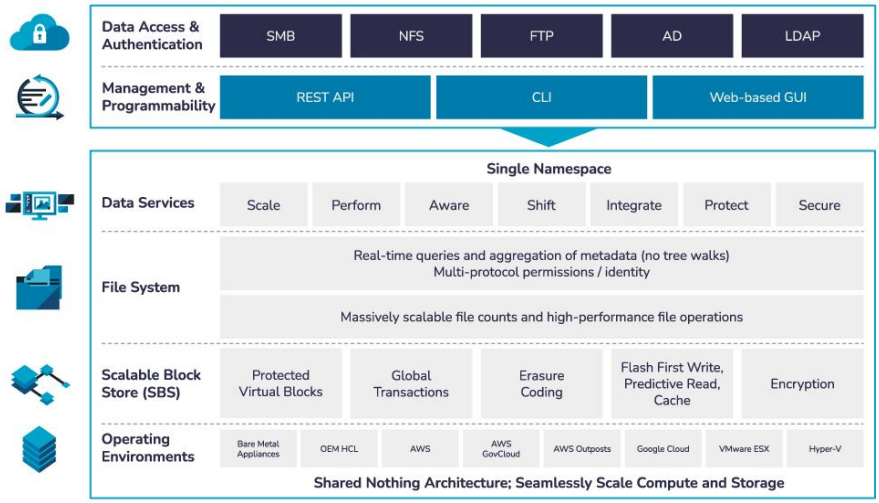

Figure 1. Qumulo File Data Platform Software Architecture

For a detailed description of the Qumulo software architecture please refer to the Software Architecture Overview.

Software Development Process

Security of our service starts, fundamentally, with the development process and at access to the source code that underpins Qumulo on Azure. The Qumulo File Data Platform is built to minimize the risk surface of an attack efficiently. The following subsections describe key preventive elements.

Locked Down Source Code

Qumulo has a central source code repository managed by our IT staff who ensure that only authorized developers can modify the source code via their SSH keys. Our code review system ensures that all code that is committed to our central repository has been vetted by at least one other developer at Qumulo before being committed.

Continuous Testing

Once code has been committed, our continuous build and test system picks it up and produces build artifacts, which are stored on a NAS. Build artifacts include both the disk images for our Qumulo Core product as well as deployable packages containing the code that runs the Qumulo on Azure service. These official build artifacts are only mutable by the continuous build system itself, select members of the IT staff and internal dev tools team. Access to the infrastructure which runs our continuous build system is controlled via per-user SSH keys.

Bi-Weekly Release Cycle

Qumulo's upgrade process is very simple and thus allows us to ship new official release candidate builds of Qumulo Core on a regular cadence, usually bi-weekly. Updates include security updates for Qumulo as well as the underlying OS, and not only allows for rapid innovation but also improved security.

Internal Scanning

Release candidate disk images undergo CIS and Qualys internal vulnerability scans before release, which is recognized by the security community as one of the best vulnerability assessment tools.

Secure coding practices

Qumulo software is developed following secure coding best practices to deliberately reduce the overall security risk, the risk of exploitable vulnerabilities, and the possibility of access to data.

Vulnerability Management

Qumulo ensures that the latest security updates for the underlying OS packages will be evaluated and resolved in a manner that reflects the impact to our customer. Remotely exploitable vulnerabilities will be prioritized and evaluated with action plans established based on CVE/CVSS levels. Remediations will be released within the appropriate timeframes based on the issue severity.

Locked-down Linux version

Qumulo is a software-only file system, which is built to run on a long term support version of Ubuntu Linux. The underlying Linux operating system is configured in order to allow only operations needed to perform the tasks

of the file system. Many other standard Linux services are disabled to reduce the attack surface.

User space application

Qumulo's File Data Platform is completely built as a user-space application. This has many advantages, those related to security are as follows:

- **No direct data access on Qumulo nodes.** Even if an attacker acquired local root user privileges and access, there is no way to access your data on the nodes directly. This is different from other storage vendors' implementations where an admin user on the nodes can access and manipulate all data locally. Accessing data through SMB or NFS would require additional software to be installed on the node. (API data access might still be possible with an access token for API data access for non-root users).
- **Fully developed native protocol stack:** There are no third-party or open source components used (such as Ganesha, Samba, or the like) to implement the data access protocols. Qumulo develops and controls every single line of code for all data access protocols. Attack surfaces due to third-party components are significantly reduced since their use is excluded.
- **Level of separation from OS:** No means of sharing users or privileges with the underlying OS. Users in the underlying Linux OS are "unknown" to Qumulo. Service user accounts are typically maintained in Active Directory or a local database but are not shared with the underlying OS.

Architected for Security

To ensure security of customer data and applications service, individual Azure subscriptions leverage the security features provided by the underlying Azure infrastructure and system architecture. In addition, Qumulo constantly looks to improve security by applying new security features as they become available.

The Service has in place various procedural, administrative, technical, and physical safeguards to help protect subscriber accounts, Qumulo environments, and data from loss, theft, misuse, abuse and unauthorized access, disclosure, alteration, and destruction.

Customer File System Subscription

The Customer File System Subscriptions contain the infrastructure that runs each customer's Service environment. To ensure complete isolation of customer data, each file system lives in its own resource group and has

dedicated infrastructure upon which Qumulo Core is running, including Storage Accounts, Network Security Groups (NSGs), Virtual Machines (VMs), and a Private Link service that allows Qumulo to access the file system for support and maintenance.

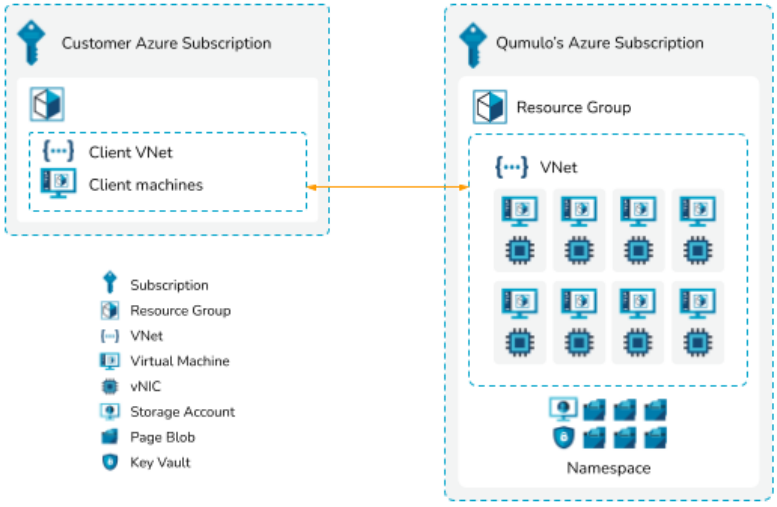

Figure 2. Qumulo on Azure Customer File System Overview

Networking

Each file system instance has an associated File System Virtual Network (VNet) in which the network interfaces for the VMs that run the Qumulo Core software reside. The File System VNet has a customer-assigned address space and will typically be peered with one or more VNets in the customer's own Azure infrastructure. VNet peering allows for private communication between VNets wherein traffic flowing across the peering is routed only through Azure's secure backbone network.

The customer is allowed Azure RBAC permission to peer the File System VNet with their own VNets by invitation into Qumulo's Azure AD tenant. VNet peering allows two-way connectivity between the File System VNet and the customer's own VNet(s), which is essential for certain protocols such as Kerberos, NSM, and FTP. Peering allows full bandwidth between VNets with all traffic staying on Microsoft's own private backbone.

Qumulo uses Azure's Network Security Group (NSG) facility to ensure that the VMs running Qumulo Core cannot be reached over the public internet nor can they, in general, initiate connections to the public internet. The two exceptions are public Azure services (via the AzureCloud service tag) and Qumulo's cloud-based monitoring service which allows our Customer

Success team to observe statistical information about the usage and performance of the file system.

In order for Qumulo to be able to access the running Qumulo Core software for operational and maintenance purposes, an Azure Private Link Service is also created in the VNet which allows unidirectional connectivity from Qumulo's Customer Success Engineers into the per-file system VNet via Azure Private Endpoints. Infrastructure running in the per-file system VNet cannot make connections into Qumulo's service networks.

Persistent Data Storage

Data stored in Qumulo on Azure file systems is stored on Azure blobs associated with Storage Accounts that are allocated per-filesystem. Qumulo leverages Azure RBAC to ensure that the VMs running any given instance of Qumulo Core, based upon system-assigned identities, only have permission to access the blobs in the storage account provisioned for that specific instance. This ensures that even if these VMs are compromised in some manner, a malicious user could not access another customer's data.

Only the VMs running the Qumulo Core software for a given file system instance and a select set of Qumulo employees (see the File System Deployer role, below) have permission to read or modify the data in these storage accounts. With approval from an affected customer, Qumulo may authorize employees in that role to directly access the data stored in the page blobs in exceptional circumstances such as file system corruption. In such cases, all access would be logged and associated with an open support ticket.

Storage accounts are locked on the back end for added protection from any possible deletion. For a storage account to be deleted that lock must first be removed.

Secure Key Vault

Access by Qumulo employees to a customer's data within the dedicated storage account is only possible after obtaining credentials from an Azure Key Vault. Access to each customer's key vault is strictly limited and any access to the key vault is logged, audited, and reviewed.

At-Rest Data Encryption

Azure blobs used for storing customer data are configured with Azure's 256-bit, FIPS 140-2 compliant encryption at rest which ensures that no customer data is ever written to persistent storage in plaintext.

In-Transit Encryption

Any connection that is controlled by Qumulo staff will be encrypted including Qumulo Core which encrypts communications with external

monitoring services (when allowed by customer) and when reading from and writing to Azure blob storage.

Connection configurations that are controlled by the customer can possibly use non-encrypted connections, but this is not recommended.

Support and Maintenance

When a customer is having a problem, Qumulo support technicians may access the VMs which are running Qumulo Core in order to assist in debugging the problem and may potentially make modifications to patch the issue. Access to these VMs is permitted via SSH from Qumulo's internal corporate network via the private endpoint mentioned above in the Networking section. SSH keys are stored in per-cluster Azure Key Vaults which utilize Azure RBAC to ensure only a limited set of employees at Qumulo can access the keys. Access of these keys is audited and stored durably in an Azure Log Analytics Workspace, and when reading from and writing to Azure blob storage. Key vault access is reviewed regularly.

Qumulo Services Operational Roles

Qumulo support services utilize Azure RBAC to manage access to infrastructure running in the customer file system Subscriptions. To ensure that only required access is granted to individual employees and services, Qumulo defines a number of operational roles which are assigned using Azure Active Directory. Roles are reviewed regularly to ensure proper access is maintained.

File system subscription operational roles are described below. All permissions are reviewed regularly:

Role	Overview	Assignees
Owner	Owners have permission to create and manage any resource in the subscription.	Select IT staff who administer Qumulo's Azure subscriptions.
File System Deployer	The File System Deployer role has permission to create, modify, and delete the actual infrastructure that runs each Qumulo on Azure file system instance. This includes access to customer data stored on Azure blobs.	Select Qumulo engineering staff who are responsible for the file system creation and maintenance tasks.
Site Reliability Engineer	Site Reliability Engineer (SRE) role has permission to read the configuration information within customer file system Subscriptions and can open support tickets with Microsoft for issues. SREs do not have access to customer data. allows for a Site Reliability Engineer to debug issues with deployed customer file system.	Select Qumulo engineering staff who may need to diagnose issues with the Qumulo on Azure service.
Customer Success Engineer	The Customer Success Engineer (CSE) role has permission to retrieve the keys necessary to SSH into the VMs running Qumulo Core within customer file system subscriptions. Once SSH'd into the VMs, these users have administrator privileges on the guest OS and can access the Qumulo REST API as the Qumulo admin user on the customer's behalf to provide support.	Select Qumulo Customer Success Engineers

VNet Peering Both Ways	Allows an assignee to peer a File System VNet provisioned by Qumulo on Azure with VNets in their own subscriptions. Role assignments are scoped to just the relevant File System VNet for the assignee.	Customer guest accounts invited into the Qumulo Azure AD tenant

Monitoring

Qumulo on Azure file systems utilize the same Cloud-Based Monitoring service included with other Qumulo file systems. In order to ensure that issues are detected swiftly and with ample information available for investigation, cloud-based monitoring is required to be enabled for Qumulo on Azure file systems. Qumulo's cloud-based monitoring service collects only the following information:

- File system name
- Information about the infrastructure upon which the file system is running
- Performance and capacity statistics
- Configuration data including metrics
- Logs, stack traces, and core dumps

NOTE The Cloud-Based Monitoring service does **not** collect file & path names, client IP addresses, and login information (such as usernames & passwords).

Alerting

Alerts will be triggered for events that occur to customer environments from which our cloud services team can more quickly troubleshoot possible issues. Some of the more critical events will also trigger alerts to the Qumulo support teams. Examples of these include:

- Namespace offline
- High CPU usage
- Low capacity alerts

Azure Status

Overall Azure health status can be found at https://status.azure.com/

Incident Management and HA

Highly Available and Redundant Infrastructure

Our Service environment is built on redundant and resilient infrastructure, designed to maintain high levels of availability. Production environments feature Qumulo Core's highly available architecture to ensure that failure of a single node will not affect production availability.

The Service uptime and support service level agreements (SLAs) can be found here:
https://qumulo.com/wp-content/uploads/2021/03/SaaS-Subscription-Service-Level-Agreement.pdf

Incident Management

While reasonable precautions are taken to secure Service environments from security threats and breaches, in any connected environment there is always a risk of security incidents that might originate from external or internal threats. The Service has in place certain teams, policies, and procedures to deal with security incidents.

Security incidents that are not automatically detected by Cloud Operations can be reported through the normal support channels, or in case of emergency contact the Qumulo Security Team at Security@qumulo.com.

INCIDENT RESPONSE PLAN

In the unlikely event of a security-related incident or breach, Qumulo has a system to report, contain, analyze, communicate, and resolve security related incidents. This incident response plan outlines the roles and procedures in place for responding to security incidents involving the Service, infrastructure and systems. An annual incident response test or table-top exercise is conducted at least annually. The plan does not cover security breaches within a customer's internal environment, or other third-party environments connected or integrated into the Service.

Monitoring

Qumulo Cloud Operations actively monitors automated metrics for system level events and will investigate and report incidents accordingly. Service penetration tests are performed periodically and identified issues are addressed.

Customers are encouraged to monitor for any unusual activity or behavior and report any suspicious or malicious events immediately by contacting Customer Success or emailing Security@qumulo.com.

Incident Reporting and Escalation

Once reported, Cloud Operations will start primary investigations. If the primary investigation warrants escalation, the incident will be escalated to a Security Incident Lead. Following investigation, if the incident is a valid security incident the Security Team is notified and assists in the incident response.

Containment

The Security Team, Cloud Operations, and Qumulo IT will initiate an immediate lock-down procedure to contain the incident and preserve any forensic evidence. The Security Incident Lead will oversee the containment process and notify Management of the incident. Customer Success will notify subscribers of any planned downtime due to lockdown and containment procedures. If additional help is required, outside forensic assistance may be utilized to assist in the investigation.

Analysis

The Security Incident Lead will coordinate with all involved parties to analyze the extent of the incident and will coordinate with executive leadership to analyze the financial and material impact of the incident. The Security Team, Engineering Leadership, and Cloud Teams will work together to determine the scope of the incident and how Service business continuity may be affected.

Communication

The Security Incident Lead will work with the General Counsel and other internal resources to involve outside authorities if required and will coordinate timely communication with customers regarding the incident and expected business continuity disruption.

Resolution

The Qumulo Security Team in conjunction with the Qumulo on Azure Engineering Leadership Team will determine next steps to resolution and if any Service change requests are needed.

Product Security Capabilities

The Qumulo Core software includes security features designed for end user administrators and users to control the use of the service. It's important for each customer to understand these Service capabilities and follow best practices for security of their Service implementation and use.

Role-based access control

Role-based access control (RBAC) allows admins to assign fine-grained privileges to regular users or groups and to alleviate their privileges where needed while keeping them as minimal as possible. This allows delegating tasks in a secure way away from the admin. Together with the Qumulo Auditing feature, it allows us to deploy a very controlled and secure management framework.

There are currently three predefined roles:

- **Administrators:** Qumulo Administrators will have full access and control of the cluster.
- **Data-Administrators:** The Data-Administrators role is ideal for API/CLI users. With this role, a user or group will not have access to the Web UI but will have the same file privileges as the Administrators role along with some others.
- **Observers:** With the Observers role, a user or group will have the privilege to access the Web UI and read-only APIs with a few exceptions (debug APIs and authentication settings).

For more detailed information on Qumulo's RBAC functionality.

Quotas

Qumulo supports Intelligent Quotas which ensure that every quota is a policy that executes a set of real-time queries. Intelligent quotas can be enforced immediately, unlike traditional systems that require tree walking of the entire directory structure and can take days to complete. The benefits of this approach include real-time diagnosis and enforcement of rogue applications and users, along with real-time visibility showing how the storage is allocated at any given point in time.

For more detailed information on Qumulo Quotas.

Hiding SMB Shares from Unauthorized Users

Qumulo allows hiding SMB Shares from unauthorized users. Mounting the share requires explicit knowledge of the share path to block potential intruders from browsing shares.

In addition, access-based enumeration can be enabled for every share. By doing so, only the files and folders that a user has permission to access will be displayed to that user. If a user does not have read or equivalent permissions for a folder, the folder is hidden from the user's view.

Encrypted Connections

While the customer can control which protocols are utilized to connect to their own systems when reading or writing data to their Qumulo on Azure file system, Qumulo recommends taking advantage of protocols (e.g. SMB3 and FTPS) which allow for in-flight encryption.

Datacenter Regions

The Qumulo on Azure Service is available to customers that can be accommodated via the following Azure datacenter regions:

Datacenter Region	Support Country Location
US East	United States
US West	United States
US East 2	United States
US West 2	United States
South Central US	United States
North Central US	United States
Central US	United States
West Central US	United States

NOTE Customers are responsible for validating that they are able to legally operate in the third-party datacenter regions described above. Customers should also be aware that in some situations support services may be hosted in a country other than where the datacenter is located.