Cloud Native Qumulo on AZURE

Reference Architecture Document for Securing CNQ Infrastructure and for Backup of Customer Data

By Paul Johnson
Senior Director, Sales Engineering

and

Justin Botticelli
Senior Systems Engineer

Qumulo

Disclaimer

The architectural recommendations and best practice guidelines contained in this document involve integration with third-party ecosystems, specifically Microsoft® Azure®. All registered trademarks, product names, and associated logos, including "Microsoft" and "Azure, remain the exclusive property of their respective trademark holders. Reference to these entities is provided strictly for technical description and informational context; such usage does not constitute an official affiliation, sponsorship, or endorsement by Microsoft Corporation.

These architectural principles are derived from operational field experience and public documentation current at the time of writing. Because Azure service configurations, API behaviors, and platform recommendations are subject to modification by Microsoft without prior notification, Qumulo, Inc. provides this technical reference **on an "AS IS" basis without warranties of any kind**. Final responsibility for architectural validation, regulatory compliance, and alignment with authoritative Microsoft implementation standards rests solely with the customer user.

Table of Contents

1. Executive Summary

Cloud Native Qumulo (CNQ) on Azure utilizes a fully disaggregated, cloud-native architecture that decouples file system compute from long-term storage. At the stateless compute layer, virtual machines leverage local read/write caching (such as NVMe and system memory) to mask underlying latencies and deliver predictable, high-throughput I/O. This compute tier operates above a persistent, elastically scalable data tier housed directly in Azure Blob Storage accounts. Because this persistent tier stores data in Qumulo’s proprietary object format, Azure Blob Storage cannot serve as an independently restorable backup boundary, and Azure-native features such as blob versioning are not supported. Consequently, data protection and recovery must be delivered directly within the Qumulo data plane—using native Qumulo Snapshots or supported third-party backup solutions—while surrounding infrastructure security is maintained using native Azure governance controls. This document describes this backup and recovery methodology in detail.

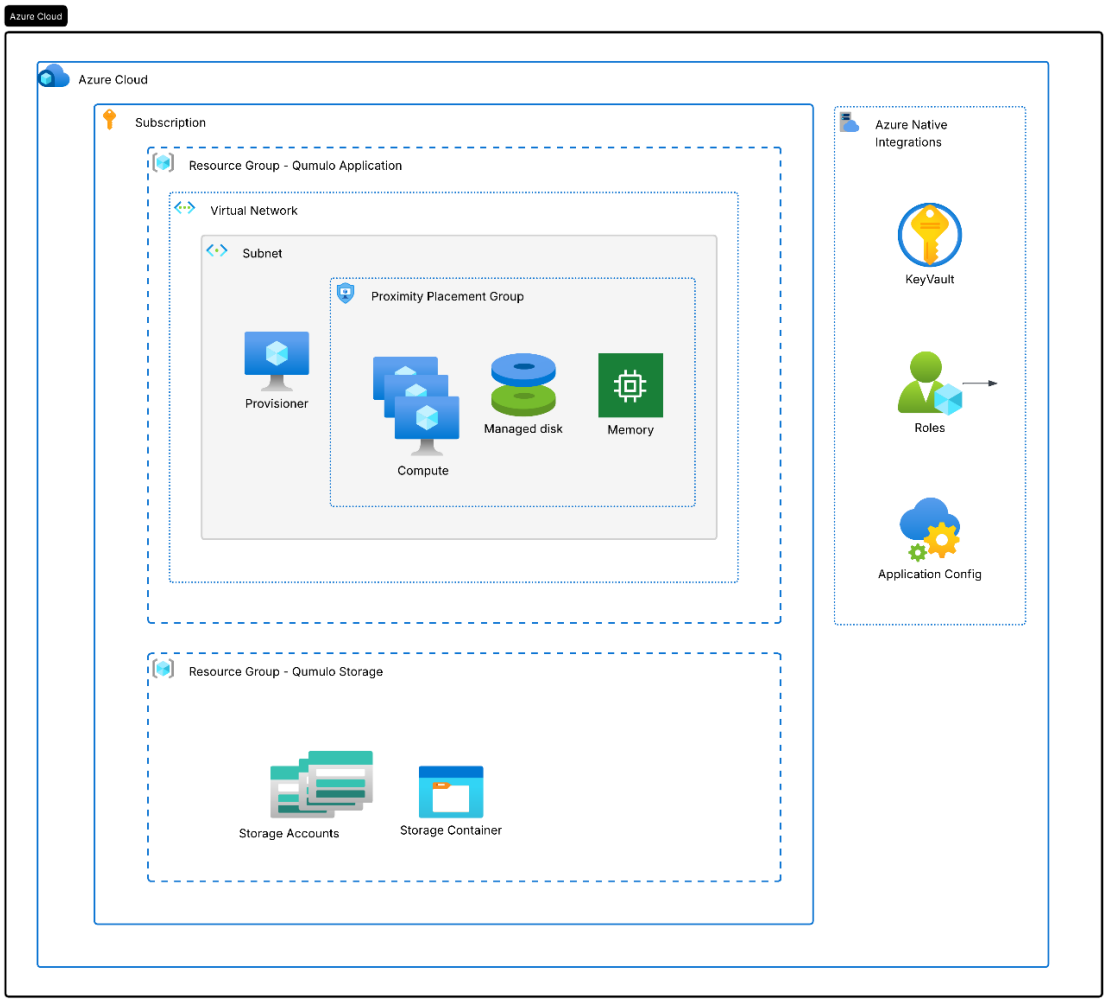

Figure 1: Architectural Diagram - CNQ on Azure

2. Azure Data Durability Framework

By leveraging Microsoft's cloud-scale infrastructure for long-term storage, CNQ on Azure redefines data integrity management, delivering up to 12 nines (99.9999999999%) of durability[1].

Azure Storage Redundancy Options

The redundancy level of the underlying Azure Blob tier can be selected based on specific cost profiles and business continuity requirements:

- **Locally-Redundant Storage (LRS):** maintaining three synchronous data copies within a single data center facility. This provides 11 nines of durability and is suitable for deep archival or colder storage workloads.

[1] Microsoft. (2024). Data Redundancy - Azure Storage. In learn.microsoft.com. https://learn.microsoft.com/en-us/azure/storage/common/storage-redundancy

- **Zone-Redundant Storage (ZRS):** replicating data synchronously across three separate physical Availability Zones (AZs) within an Azure region. This increases durability to 12 nines, ensuring zero data loss even during a catastrophic facility failure. For highly regulated environments, the recommended architecture utilizes ZRS, configured via Terraform at deployment. Availability of ZRS varies by region.

Qumulo utilizes an authoritative, fully journaled metadata layer rather than relying on the eventual consistency common in other cloud file systems. Strict consistency is enforced; every file modification or write is committed across all compute nodes immediately. Because of this architectural requirement, native **Azure Blob snapshots would result in data inconsistency and are not supported.** Data protection is instead delivered via the methods discussed in Section 3.

Because the persistent tier absorbs all data and metadata commits, the compute nodes function as stateless entities.

In the event of a hypervisor crash or hardware fault, the system avoids time-consuming cryptographic rebuilds or RAID parity calculations. Instead, surviving nodes simply map the transaction logs from the shared persistent tier to restore multi-protocol access in seconds.

2.1 Qumulo Design Principles

- Defense in depth: independent, layered controls for the data and the infrastructure.
- Immutability at the data layer: locked (WORM-equivalent) Qumulo snapshots.
- Off-cluster resilience: continuous replication to a second cluster in a separate subscription/region or on premise data center cluster.
- Least privilege and separation of duties: no single administrator can destroy data or storage. https://learn.microsoft.com/en-us/azure/azure-resource-manager/management/lock-resources?tabs=json
- Prefer native, supported controls over bolt-on mechanisms that are not mutually compatible
- Recoverability is validated, not assumed: recovery paths are tested on a defined cadence.

2.2 Recommended Policies at a Glance

- Data protection: locked immutable Qumulo snapshots + continuous replication to a DR cluster.
- Storage efficiency: incremental-forever replication (only changed data), not full blob copies.
- Identity: least-privilege managed identities; privileged roles are PIM-eligible with approval.
- Deletion protection: resource-group **CanNotDelete** lock plus a management-group deny policy.

- Encryption: customer-managed keys in Azure Key Vault (soft-delete + purge protection enabled).
- Network: private endpoints, no public network access, restricted ports, Qumulo authoritative DNS.

WARNING Explicit non-goal: Azure Blob-level backup, versioning or WORM immutability MUST NOT be applied to the CNQ storage accounts. Qumulo owns and continuously rewrites those objects; applying Blob-tier data-protection features would break or bloat the object store. Data immutability is provided by Qumulo locked snapshots instead. It would also be necessary to have all storage accounts blobs perfectly synchronized because of the proprietary layout of the Qumulo data in the Blob store, something Azure backup does not guarantee, and why this is not a supported method of backing up Qumulo infrastructure.

3. CNQ on Azure Architecture & Data-Protection Model

3.1 Architecture Overview

A CNQ on Azure cluster is deployed with Terraform in two layers. The persistent-storage layer provisions multiple Azure Blob storage accounts (plus an Azure Key Vault and supporting configuration). The compute layer provisions the Qumulo VMs and their cache disks (read cache, write cache, and DKV) and installs Qumulo Core onto the base OS image. The two layers are independent by design: compute can be replaced while the persistent storage is retained.

```
MANAGEMENT GROUP  (governance boundary)
  policy: Azure Policy denyAction on delete  ·  PIM  ·  least-privilege RBAC
  │
  ├─ SUBSCRIPTION A — PRIMARY   (VNet: private, no public inbound)
  │     ├─ CNQ compute nodes (read / write cache tier) — private endpoint
  │     ├─ Persistent storage — Blob accounts + Key Vault (CMK, private EP)
  │     │       • resource-group CanNotDelete lock
  │     │       • locked (immutable) Qumulo snapshots  ← primary data protection
  │     └─ replication  ──async, incremental──▶  Subscription B DR cluster
  │
  └─ SUBSCRIPTION B — DR / BACKUP   (VNet: private)
        └─ CNQ cluster (DR target) — Blob accounts
```

Figure 2: CNQ on Azure trust boundaries and data-protection topology

3.2 Why the Persistent (Blob) Tier Is Not a Backup Boundary

- The blob objects are in Qumulo's proprietary object format and are not independently readable. The blob objects are highly optimized, compressed and encrypted Blob objects; therefore they are not readable by standard Azure methods. CNQ compression and encryption allows for a heightened security posture and a saving on storage costs.

- There is no supported procedure to stand up a cluster from a copy/restore of the storage accounts. *Note: this would be a deep engineering task and not a guaranteed recovery mechanism.*
- Client-acknowledged writes may reside in the compute-node cache tier (which has its own protection mechanism across Azure Managed Disk); the blob tier alone is not guaranteed to be complete without the cache tier, unless all flushes are complete with a cluster shutdown.
- Coordinating a consistent point-in-time image across multiple independent storage accounts is not something the Azure platform provides atomically.

For these reasons, backup and point-in-time recovery are implemented at the Qumulo layer, which produces consistent images by design and restores into a usable file system.

3.3 Data-Protection Model (Layered)

The stack below reads top-to-bottom, client access down to the Azure persistent tier, and shows which layer owns data protection at each level.

Layer (top → bottom)	Data-Protection Responsibility
Client access — SMB / NFS / S3	Applications read and write; no data-protection role.
Qumulo Core file system	Locked (immutable) snapshots — point-in-time, WORM-equivalent. Primary data protection.
Continuous replication	Async incremental-forever copy to a second Qumulo cluster in a separate subscription / DR.
Azure persistent tier — Blob accounts (+ CMK)	Guarded by least-privilege RBAC, resource lock, and deny policy. NOT Azure blob WORM / versioning.

Table 1: Data-protection responsibilities by layer

4. Backup & Recovery Reference Architecture

4.1 Immutable (Locked) Snapshots

Qumulo Core snapshots are instant, storage-efficient, point-in-time images of the file system. 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. Locked snapshots are the primary defense against accidental deletion, malicious deletion, and ransomware.

- Scheduled locked snapshots provide frequent recovery points (cadence per RPO target below).
- Retention is policy-driven; locked snapshots cannot be removed before expiry.
- End users recover files directly from the read-only .snapshot directory over SMB/NFS.

4.2 Off-Cluster Copy — Continuous Replication option

Continuous replication asynchronously copies data to a second Qumulo cluster and produces a consistent point-in-time copy on the target. Replication is incremental-forever — after the initial baseline, only changed data is transferred — which is the storage- and bandwidth-efficient way to protect large (multi-hundred-TB to PB) datasets. The target cluster should reside in a separate subscription (ideally a separate tenant) so that a compromise of the primary's Azure control plane does not impact the second copy of the data.

4.3 Recovery Objectives and Scenarios

Scenario	Primary Mechanism	RPO	RTO
Accidental file deletion / overwrite	Locked snapshot (.snapshot restore)	Last snapshot	Minutes
Data corruption / ransomware	Restore from locked snapshot	Last snapshot	Minutes–hours
Compute / VM loss (storage intact)	Cluster replacement on existing storage	Zero (no data loss)	Hours
Loss of primary cluster or subscription	Fail over to DR replica cluster	Last replication	Hours

Table 2: RPO/RTO values above are generic placeholders to be ratified by the board against business requirements; the architecture supports tightening them by increasing snapshot and replication frequency

4.4 Recovery Playbooks (Summary)

- File-level: browse the .snapshot tree and copy the required version back into place.
- Data loss/corruption: restore the affected directory or file system from the newest clean locked snapshot.
- Compute loss: redeploy compute in a new Terraform workspace, in the existing subscription, pointed at the existing persistent storage (cluster-replacement flow); no data is moved.
- Site/subscription loss: promote the DR replica cluster and redirect clients; re-establish replication in the reverse direction when the primary returns.

4.5 Third-Party Backup Integration (Complementary Layer)

Locked snapshots and continuous replication are the primary, recommended protection for CNQ. A third-party backup application is a complementary layer that produces an independent copy in a separate backup format and vendor domain — useful for long-term

retention, air-gapped or logically isolated copies, and satisfying a policy that requires a distinct backup product. It does not replace snapshots and replication. Note that Azure-native Azure Backup does NOT protect the Qumulo file system — it targets Azure Files, Azure VMs, and Blob, not an external NAS — so the Qumulo file system is protected over its standard SMB / NFS / S3 client protocols by a backup product that runs on an Azure access node (backup proxy/gateway) with network reachability to the cluster.

Where supported, backup products use Qumulo's SnapDiff APIs to avoid full tree walks: a snapshot and change-tracking query returns the delta between two snapshots so incrementals scale to billions of files. This requires the access node to reach the cluster on the Qumulo change-tracking API port (TCP 8000) in addition to the data protocol; if that path is blocked, incrementals fall back to slow full file-system scans. Use a least-privilege service account for the backup product and land backups in the backup vendor's storage domain, not back onto the same Qumulo cluster.

Backup Product	Integration with Qumulo	Notes
Commvault (CommCell / Metallic SaaS)	Native "Qumulo File Storage" integration using Qumulo snapshot + Change File Tracking (CFT) APIs	Fast incrementals via CFT; needs inbound TCP 8000 from the access node; Metallic needs a backup gateway near the cluster; backups land in Metallic/BYOS storage.
Cohesity	Change-tracking (SnapDiff-style) API for incrementals	NAS-mode protection over NFS/SMB with fast incremental via change tracking.
Veeam / Rubrik	Standard SMB / NFS (Veeam also supports S3 object)	Protect as a generic NAS share/object target; no confirmed deep CFT/SnapDiff integration.

Table 3: Field guidance, not a formal Qumulo support matrix — confirm the current certified version and configuration with the backup vendor and Qumulo before committing a design

Where possible, Qumulo recommends leveraging one of the above vendors, since the integration with Qumulo SnapDiff APIs does reduce the networking and API-related costs of the backup operations.

5. Ransomware Protection of Qumulo Filesystem Data

5.1 NeuralProtect Overview

Contemporary cloud infrastructures are increasingly vulnerable to advanced zero-day ransomware that deliberately evades legacy perimeter security and targets backup repositories. Conventional data protection strategies, which often depend on reactive entropy analysis or infrequent backup cycles, fail to achieve the necessary recovery point objectives (RPO), leaving organizations exposed to significant operational impact. **Qumulo NeuralProtect™** mitigates these risks by providing inline, real-time threat identification and automated response at the data-plane layer. Utilizing Elastio's AI-driven engine, NeuralProtect enables continuous, low-latency inspection across hybrid environments—

including Azure, AWS, and GCP—by performing out-of-band content analysis (e.g., 64 kB sampling) to ensure file system performance remains unaffected. The integrated engine employs deterministic models for over 2,000 malware families, temporal analysis to identify stealthy encryption patterns, and statistical AI to maintain a false-positive rate below 0.01% for zero-day detection.

5.2 Technical Architecture & Automated Threat Mitigation

Deeply integrated with **Qumulo Core**, NeuralProtect operationalizes an automated **Detect, Isolate, and Recover** framework to address security events as they occur. When the system identifies anomalous behavior or signature matches, it executes programmatic response playbooks via REST APIs. These automated defensive actions include the generation of immediate file-system snapshots, the isolation of corrupted objects within the /.qumulo_reserved/quarantine directory, and the revocation of compromised client sessions. Furthermore, critical alerts are streamed to enterprise SIEM platforms for centralized auditing. Administrators gain comprehensive visibility through the **Qumulo NEXUS Threat Detection Dashboard**, which supports granular policy enforcement and proactive protection settings. By pairing predictive AI with instant, uncorrupted recovery, NeuralProtect ensures a resilient architecture against sophisticated malware threats.

5.3 Ransomware mitigations recommendation

The threat of ransomware impact cannot be mitigated with a single solution. It has to be a combination of end point protection, immutable snapshots and backups, along with a dedicated Ransomware protection software monitoring the file data using a native solution like Qumulo NeuralProtect.

6. Securing the Azure CNQ Environment (Customer Controlled)

6.1 Identity & Access (Least Privilege)

- The cluster runs under least-privilege user-assigned managed identities (compute and provisioner), each scoped to purpose-built custom roles rather than broad built-in roles.
- Standing Owner and User Access Administrator assignments are minimized or eliminated.
- Privileged roles are made PIM-eligible with just-in-time activation and required approval, enforcing separation of duties (a second, designated approver must authorize elevation).

6.2 Deletion Protection — “No Single Administrator”

A **CanNotDelete** lock protects the storage resource from control-plane deletion, but it can be removed by any holder of Owner or User Access Administrator. It is therefore necessary but not sufficient against a privileged insider.

The architecture layers two additional controls so no single administrator can unilaterally destroy the storage.

- Resource-group **CanNotDelete** lock on the persistent-storage resource group (accidental-deletion guard).
- A management-group Azure Policy **denyAction** on delete — a system-protected control enforced above the subscription, which a subscription Owner cannot remove. This policy should be applied to the persistent storage accounts or resource group.
- PIM approval (Section 4.1) ensures elevation to a delete-capable role requires a second party.

WARNING: **ReadOnly** ***locks must never be used on CNQ resource groups***. The **azurerm** provider issues **listKeys** during plan and refresh in both Terraform workspaces and **ReadOnly** blocks that call, so a **ReadOnly** lock breaks all Terraform operations, not just deletes.

6.2.1 Terraform Compatibility

The lock is implementable natively by setting `az_lock_rg = true` in the persistent-storage workspace (note the variable defaults to false even though the **tfvars** comment claims true), and no post-deployment operation, scale up, two step scale down, or cluster replacement, deletes anything in the persistent storage resource group, so the lock never conflicts with Terraform. Removing the lock requires Owner, User Access Administrator, or a custom locks role; the Contributor role cannot remove this lock.

Operationalizing the "No Single Administrator" Deletion Guard

To ensure no individual identity possesses the authority to unilaterally dismantle storage protections, three distinct governance layers must be integrated:

- **Elimination of standing privilege:** Utilizing Microsoft Entra PIM, the Owner and User Access Administrator roles are configured as eligible rather than active. This ensures administrators hold no permanent destructive permissions.
- **Elevation via independent approval:** PIM must be configured to mandate authorization from a separate, designated party. Self-approval is blocked by PIM, enforcing a mandatory two-person control for any privilege activation.
- **Enforcement at management-group scope:** An Azure Policy **denyAction** (or Deployment Stack deny assignment) is applied at the management group above the subscription. These assignments are system-protected; because the control resides above the subscription boundary, even a subscription Owner lacks the authority to remove it.

In summary: no lone administrator can maintain standing delete rights, self-approve an elevation request, or strip a deny policy that is governed by a management group beyond their control.

Securing the CNQ Data Plane Against Unauthorized Deletion

An additional, critical security gap at the Azure Blob level must also be addressed:

- Standard **CanNotDelete** locks only protect the control plane and do not inhibit data-plane blob deletions. Any identity with Storage Blob Data Contributor rights or access to an account key can still delete objects resulting in data loss at the filesystem level.

6.3 Data-Plane Protection

- Keep shared key access enabled and protect the key material. Private endpoints with public access disabled, Key Vault RBAC restricted so only node identities can read the SAS definitions, periodic key regeneration through Key Vault, and audit alerts on **listKeys** and SAS reads.
- Do NOT enable Azure blob soft-delete, versioning, or WORM immutability on CNQ storage accounts — these interfere with Qumulo's object management. Data immutability is delivered by Qumulo locked snapshots instead.

6.4 Encryption

- Customer-managed keys (CMK): an RSA-2048/HSM key in Azure Key Vault encrypts all blob data and managed disks via a disk encryption set. Key Vault must use the Azure RBAC permission model and reside in the same subscription.
- The disk encryption set is protected from accidental destruction in Terraform via lifecycle prevent_destroy.
- Protect the key itself: enable Key Vault soft-delete and purge protection and restrict Key Vault RBAC — deletion of the key is equivalent to destruction of the data.
- Qumulo Core additionally provides its own encryption-at-rest layer (local or external KMIP key).

6.5 Network Security

- Private endpoints for Blob storage, Key Vault, and App Configuration; public network access disabled.
- VNet service endpoints for Microsoft.Storage and Microsoft.KeyVault on the cluster subnet.
- Restrict NSGs to required ports only; allow the mandatory outbound endpoints for monitoring/provisioning.
- Qumulo authoritative DNS (QDNS) via Azure Private DNS Resolver for the private topology.

6.6 Logging and Auditing

To ensure the integrity of evidence and facilitate automated threat intervention, diagnostic telemetry must be isolated from the primary infrastructure control plane. Configure Azure Diagnostic Settings on the storage accounts, Key Vault, and associated network resource groups to stream all Write, Delete, and Action operations to an Azure Log Analytics Workspace or Azure Sentinel instance. This log destination must reside within a dedicated, restricted security resource group to prevent unauthorized modification or deletion by standard infrastructure administrators. Automated security playbooks should be deployed

to trigger immediate alerts upon critical administrative actions, such as modifications to network routing tables, NSG firewall rules, or storage account configurations.

7. Governance, Assumptions & Board Considerations

7.1 Threat → Control Mapping

Threat	Control(s)
Accidental deletion of storage account / RG	CanNotDelete lock + management-group deny policy
Rogue admin deletes the storage account	Deny policy at management group (Owner cannot remove) + PIM approval
Rogue admin / attacker deletes or encrypts data	Locked immutable snapshots + replication to separate subscription
Ransomware / data corruption	Locked snapshots (immutable recovery points)
Encryption-key deletion or compromise	Key Vault soft-delete + purge protection; restricted Key Vault RBAC
Loss of region / subscription / tenant	DR replica cluster in separate subscription; documented failover
Network exposure of storage / Key Vault	Private endpoints; public access disabled; least-privilege NSGs
Excessive standing privilege	Least-privilege managed identities; PIM eligible + approval

Table 4: Threats, and the accompanying control setting to prevent them

Reference Documentation

Qumulo — CNQ on Azure architecture & deployment

- How Cloud Native Qumulo on Azure Works — architecture, reference design, known limits
- Deploying CNQ on Azure with Terraform — prerequisites, persistent-storage + compute deployment, and the *cluster-replacement* procedure

Qumulo — data protection & immutability (the core of your backup story)

- How Snapshots Work in Qumulo Core
- Managing Snapshots — policies, schedules, expiration
- Locking and Unlocking Snapshots — *the supported WORM/immutability mechanism*
- Recovering Files by Using Snapshots — `.snapshot` restore
- Creating & Managing a Continuous Replication Relationship — off-cluster PIT copy to DR

Qumulo — security & operations

- Managing Encryption at Rest — software encryption + external KMIP key
- Managing RBAC
- Required Networking Ports — NSG/firewall reference
- *CNQ `azure-terraform-cnq` deployment README* (bundled in the deployment package) — the authoritative source for the Azure infra-security controls (CMK, private endpoints, resource-group lock `az_lock_rg`, least-privilege managed identities/RBAC). Not on the public docs site.

Microsoft Azure — deletion protection & governance ("no single admin")

- Lock your Azure resources (CanNotDelete) — note: control-plane only; doesn't protect data-plane
- Azure Policy denyAction effect — management-group deny-on-delete a subscription Owner can't remove
- PIM — approve requests for Azure resource roles and for Entra roles

Microsoft Azure — encryption & network

- Customer-managed keys for Azure Storage
- Server-side encryption of Azure Disk Storage
- Azure Key Vault security features
- Use Private Endpoints for Azure Storage