MinIO in Healthcare

FEBRUARY 2020

The healthcare industry is unique in so many ways.

To begin with, it lacks a common objective function on which to optimize. This is because different healthcare organizations, both payers and providers, value different things. Profitability is not the universal goal - there are teaching hospitals, faith-based hospitals and not-for-profit insurance plans. Consider also that a large portion of the industry is guided by a 2,500-year oath at the point of interaction. Further, healthcare is at once highly regulated and process-driven yet intensely personalized. The list goes on with each example underscoring how different healthcare is from other industries.

Despite the uniqueness of the industry, it does share a common challenge with every other industry - that of managing and extracting value from their data.

Healthcare produces a disproportionate share of the world's data - estimated at 30% of electronic storage. More importantly, the rate of growth is stunning. An IDC/DellEMC report found that between 2016 and 2018, data grew from 1.45 PB to 9.70 PB or 878%. Assuming that growth rate has not abated - which is conservative - those healthcare organizations would now be managing close to 100 PB each.

The volume of data and the attendant growth rate have required the healthcare industry to radically re-evaluate their storage architecture. What was a solved problem four years ago is now the primary challenge facing the industry - even larger than analytics.

The most sophisticated payers, providers and technology vendors have come to the realization that object storage, whether in the cloud or on-premises is the answer. This paper outlines why and the specific benefits of MinIO's object storage suite.

Why Object Storage is now the Foundational Storage Tier in Healthcare

The primary reason for the shift from SAN/NAS to object has to do with the aforementioned scale. Those legacy systems were not designed to perform at this scale. Further, no database can fit 100 PB, even if the data were arranged neatly as columns and rows, which it is not. Modern healthcare data is unstructured - images, scans, log data, event streams.

While the unstructured nature of data and the volume is a key driver for object storage, it is not traditional object storage that healthcare is adopting. Traditional object storage was not designed for speed or to be cloud-native or application-oriented. It was designed for archival or backup.

Amazon S3 changed that. They proved that object storage could be fast and cloud native. Furthermore, with the advent of S3 Select, AWS proved that object storage could also be searchable. This was a game changer.

Object storage could now serve as the foundational component of healthcare's storage architecture. Consider the following. Modern object storage is inserted in the list below to distinguish between traditional, archival appliance-based object storage and what is sweeping the healthcare industry:

1) Object storage scales without complexity.
2) Object storage is designed for unstructured data.
3) Object storage offers vastly superior economics vs. block or file.
4) Modern object storage API is RESTful (based on Amazon's S3 API) making it compatible with cloud native applications.
5) Modern object storage is searchable using standard SQL.

6) Modern object storage is fast. In the case of MinIO, it is hyper-fast, delivering speeds in excess of 171 GB/s on 32 nodes of NVMe and 9.4 GB/s on 24 nodes of HDD.
7) Modern object storage is cloud-native and suited for containerization and orchestration, making it truly elastic.

For these reasons and others, modern object storage has become the default storage architecture in healthcare. It will take time to migrate away from legacy applications that depend on SAN and NAS, but not as long as one might think. Those solutions cannot scale. They have to be replaced - it is a business imperative, not a technical decision.

The New Edge Model + A 360 Patient View

Healthcare data is created at the edge and shared across the organization. This is simply how it works. MRIs, CTs, X Rays, Ultrasounds are taken at the edge, are digital from inception, are shared in real time with the doctor and ultimately stored somewhere more central. Almost every single interaction from labs to IoT data from kidney dialysis machines follows the same model: data is collected or generated at the edge, made readily available to the organization and ultimately stored elsewhere.

Modern object storage is inherently built to share data efficiently across the Internet. SAN/NAS systems are designed for low latency applications within the data center. Consider the fact that even on AWS, you cannot access EBS (Elastic Block) or EFS (Elastic File) from outside the cloud. It is just not set up for that.

This modernization of the architecture has massive implications for the cost of healthcare. By moving the legacy IT infrastructure out of the individual hospitals into a common shared cloud infrastructure, all of the benefits of containerization and orchestration come into play.

By developing dense object storage clusters at the edge, the information can be processed and shared locally and eventually migrated to a lower tier, either in a data center or into the public cloud if the security and resilience meet requirements.

The Application Economy + Decline of File and Block

Healthcare is an industry defined by applications. Epic is an application. Cerner is an application. IBM Watson is an application. Every insurance company has dozens, if not hundreds, of applications - from population health to analytics. Every provider has dozens, if not hundreds, of applications from the EHR to chatbots.

Those applications require data. Modern applications communicate through RESTful APIs like Amazon's S3 API. While many of those older applications may specifically target SAN/NAS environments with POSIX, they are rapidly giving way to RESTful solutions like S3.

Specifically, applications that run on the cloud (public or private) and at the edge are, by default, using RESTful APIs. Further, they can run in the cloud but access data at the edge - a critical model in healthcare. One only needs to consider the model that dominates today, an analytics application running on the cloud but accessing data that exists in the hospital. It is simply how modern software is architected.

Providers are not building data centers at each hospital location. They are using applications (speaking S3) to engage with data that is generated at the edge. Further, as we see more convergence between payer and provider (where each adapts skills from the other), we will see more of this application-oriented architecture and therefore more object storage in both places.

One last proof point: The foundation for interoperability in healthcare, FHIR, is based on RESTful APIs.

True Security

Security is another reason that modern object storage has flourished in healthcare. Modern object storage like MinIO or AWS has advanced encryption algorithms for data in flight or at rest.

In the case of a hospital, data loss is preferred over a data breach.

This, however, is not a choice a software architect should have to, or wants to, make. They want security and resilience. Private cloud object storage can offer both. Using inline erasure coding and bitrot protection, MinIO is more resilient and more economical. More importantly, the fact that the data can remain within the control plane of the payer or provider insures against the types of breaches that have plagued the public cloud. Add to this key features like sophisticated encryption, tamper-proofing, object locking and legal lock for governance and compliance and you have a fundamentally more secure solution.

Public cloud vs. Private Cloud

We often get the question from our healthcare clients: public or private cloud? As a private cloud object storage provider our answer often surprises.

We tell our clients to start in the public cloud. There are skills to be mastered, best practices to be applied, lessons to be learned. Teams can develop expertise and adopt the mindset of the cloud. In the cloud they will learn about elasticity, containerization, Kubernetes, micro-services, best of breed for every component.

When those skills have been mastered - repatriate.

This ensures a private cloud that is based on the best practices of the industry while delivering superior resilience, security and economics.

Ultimately, data will exist everywhere. In multiple public clouds. On premises and in the public cloud. This requires a single API to speak across those disparate instances. As we have written above, that API is S3. MinIO's credentials in that space are unquestioned. It is why Google and Microsoft both asked us to write S3 to Azure/GCP gateways for their clouds.

Why MinIO

The first part of this paper focused on why object storage is the foundational storage type for healthcare. This section of the paper focuses on why MinIO is the dominant provider of private cloud object storage.

MinIO is a high-performance, software-defined, distributed object storage system. By following the methods and design philosophy of hyperscale computing providers, MinIO delivers superior performance and massive scalability to a wide variety of workloads in the private cloud.

While MinIO is ideal for traditional object storage use cases like secondary storage, disaster recovery and archiving, it truly excels in overcoming the challenges of executing high-performance computing against massive datasets.

In the modern enterprise these consist of machine learning, analytics and cloud-native application workloads.

Because MinIO is purpose-built to serve only objects, a single-layer architecture achieves all of the necessary functionality without compromise. The advantage of this design is an object server that is high-performance and lightweight.

MinIO combines the inherent advantages of object storage with a robust suite of features, a stunningly simple, intuitive interface and an expansive set of integrations.

MinIO is unique in that it was built from the ground up with cloud-native technologies to be simple, fast, durable and highly scalable. With the belief that a complex solution cannot be scalable, a minimalist design philosophy forms the foundation of the MinIO architecture design.

The result is a system that is excels across several key dimensions:

- Performance. With its focus on high performance, MinIO enables enterprises to support multiple use cases with the same platform. For example, MinIO's performance characteristics mean that you can run multiple Spark, Presto, and Hive queries, or to quickly test, train and deploy AI algorithms, without suffering a storage bottleneck. MinIO object storage is used as the primary storage for cloud native applications that require higher throughput and lower latency than traditional object storage can provide.

- Scalability. A design philosophy that "simple things scale" means that scaling starts with a single cluster which can be federated with other MinIO clusters to create a global namespace, spanning multiple data centers if needed. Gradual expansion of the namespace is possible by adding more clusters, more racks, and even by adding more data centers to the MinIO single namespace. MinIO leverages the hard won knowledge of the web scalers to bring a simple scaling model to object storage.

- Simplicity. Minimalism is a guiding design philosophy at MinIO. Simplicity reduces opportunities for errors, improves uptime, delivers reliability while serving as the foundation for performance. MinIO can be installed and configured within minutes simply by downloading a single binary and then executing. The amount of configuration options and variations is kept to a minimum which results in near-zero system administration tasks and few paths to failures. Upgrading MinIO is done with a single command which is non-disruptive and incurs zero downtime - lowering total cost of ownership.

High Performance Object Storage

Every feature of MinIO's object storage suite was designed to deliver performance and scale. As a software-defined solution, MinIO can be paired with hundreds of different compute and storage configurations from Intel Skylake or Xeon Gold processors to NVMe drives, spinning disk - even tape.

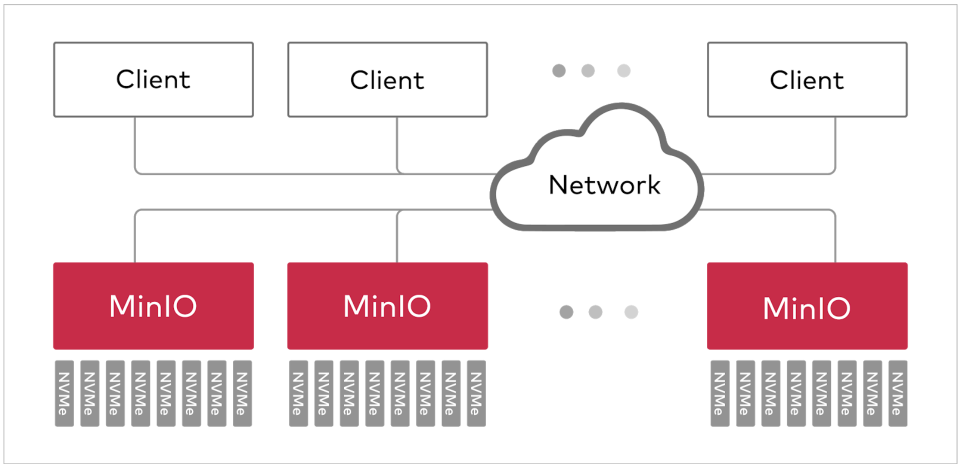


Figure 3: A typical MinIO deployment.

MinIO's software-defined object storage suite consists of a server, an optional client, and an optional software development kit (SDK):

- MinIO Server. MinIO is a distributed object storage server released under Apache License v2.0. It boasts the most comprehensive implementation of the Amazon S3 API to be found anywhere outside of Amazon itself. MinIO is feature-complete, providing enterprise-grade encryption, identity management, access control, and data protection capabilities, including erasure code and bitrot protection.

- MinIO Client. Called mc, the MinIO Client is a modern and cloud-native alternative to the familiar UNIX commands like ls, cat, cp mirror, diff, find and mv. This client provides advanced functionality that is suitable for web-scale object storage deployments. For example, powerful data replication tools work between multiple sites for HA(highly availability) and DR (disaster recovery) purposes and support generating shared, time-bound links for objects.

- MinIO SDKs. The MinIO Client SDKs provide simple APIs to access any Amazon S3-compatible object storage. MinIO repositories on Github offer SDKs for popular development languages such as Golang, JavaScript, .Net, Python and Java.

The features of MinIO's Object Server are notable for their breadth, depth and focus on the enterprise. As a cloud-native implementation, the range of features exceed those in legacy or bolt-on implementations while the attention to engineering first principles ensure exceptional performance.

S3 Select

To deliver big data, analytic and machine learning workflows requires filtered access to the data - grabbing just what is relevant to a particular job.

MinIO has developed its own implementation of the S3 Select API, which is essentially SQL query capabilities baked right into the object store. Users can execute SELECT queries on their objects, and retrieve a relevant subset of the object, instead of having to download the whole object. With the S3 Select API, applications can now download a specific subset of an object - only the subset that satisfies given SELECT query. This directly translates into efficiency and performance by reducing bandwidth requirements, optimizing compute and memory resources meaning more jobs can be run in parallel - with the same compute resources. As jobs finish faster, there is better utilization of analysts and domain experts.

Erasure Coding

MinIO protects data with per-object, inline erasure coding which is written in assembly code to deliver the highest performance possible. MinIO uses Reed-Solomon code to stripe objects into n/2 data and n/2 parity blocks - although these can be configured to any desired redundancy level. This means that in a 12 drive setup, an object is shared across as 6 data and 6 parity blocks. Even if you lose as many as 5 ((n/2)–1) drives, be it parity or data, you can still reconstruct the data reliably from the remaining drives. MinIO's implementation ensures that objects can be read or new objects written even if multiple devices are lost or unavailable.

Erasure code protects data without the high storage overhead of using RAID configurations or data replicas. For example, RAID-6 only protects against a two-drive failure whereas erasure code allows MinIO to continue to serve data even with the loss of up to 50 percent of the drives and 50 percent of the servers.

Finally, MinIO applies erasure code to individual objects, which allows the healing of one object at a time. For RAID-protected storage solutions, healing is done at the RAID volume level, which impacts the performance of every file stored on the volume until the healing is completed.

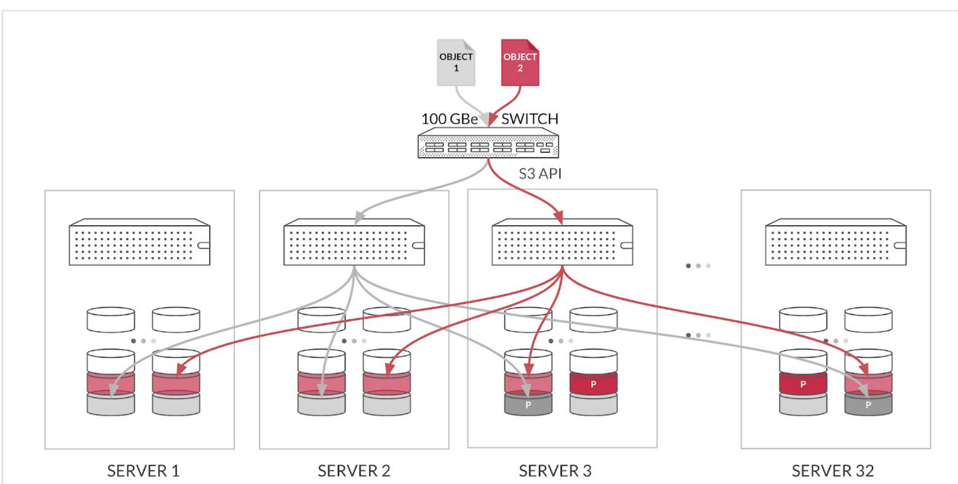


Figure 4: Erasure code protects data without the overhead associated with alternative approaches.

Bitrot Protection

Silent data corruption, or bitrot, is a serious problem faced by disk drives resulting in data getting corrupted without the user's knowledge. The reasons are manifold (aging drives, current spikes, bugs in disk firmware, phantom writes, misdirected reads/writes, driver errors, accidental overwrites) but the result is the same - compromised data.

MinIO's optimized implementation of the HighwayHash algorithm , ensures that it will never read corrupted data - it captures and heals corrupted objects on the fly. Integrity is ensured from end to end by computing hash on READ and verifying it on WRITE from the application, across the network and to the memory/drive. The implementation is designed for speed and can achieve hashing speeds over 10 GB/sec on a single core on Intel CPUs.

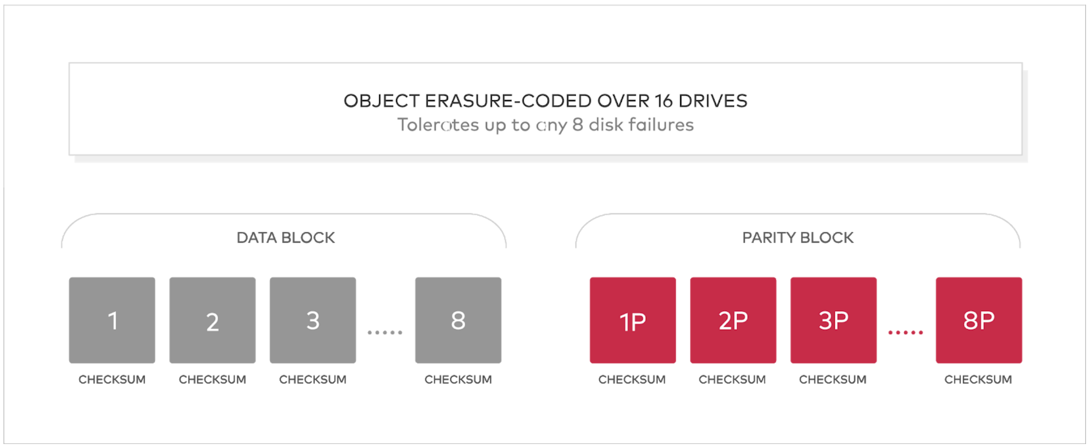


Figure 5: MinIO's data protection schemes cover failure and silent data corruption.

Identity and Access Management

MinIO supports the most advanced standards in identity management, integrating with the OpenID connect compatible providers as well as key external IDP vendors. That means that access is centralized and passwords are temporary and rotated, not stored in config files and databases. Furthermore, access policies are fine grained and highly configurable which means that supporting multi-tenant and multi-instance deployments become simple.

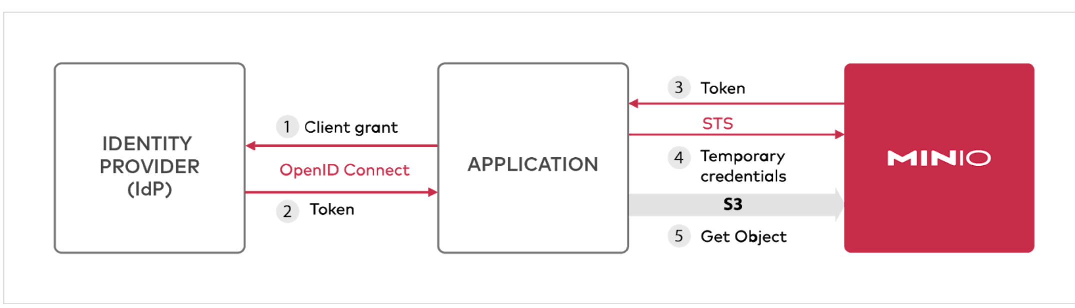


Figure 6: Identity protection and single sign on (SSO) are critical enterprise features.

Encryption and WORM

It is one thing to encrypt data in flight it is another to protect data at rest. MinIO supports multiple, sophisticated server-side encryption schemes to protect data - wherever it may be. MinIO's approach assures confidentiality, integrity and authenticity with negligible performance overhead. Server side and client side encryption are supported using AES-256-GCM, ChaCha20-Poly1305 and AES-CBC. Encrypted objects are tamper-proofed with AEAD server side encryption. Additionally, MinIO is compatible with and tested against all commonly used Key Management solutions (e.g. HashiCorp Vault).

MinIO uses key-management-systems (KMS) or cryptographic key management system (CKMS) to support SSE-S3.If a client requests SSE-S3, or auto-encryption is enabled, the MinIO server encrypts each object with a unique object key which is protected by a master key managed by the KMS. Given the exceptionally low overhead, auto-encryption can be turned on for every application and instance.

When WORM is enabled, MinIO disables all APIs that can potentially mutate the object data and metadata. The means that data once written becomes tamper-proof. This has practical applications for a number of different regulatory requirements.

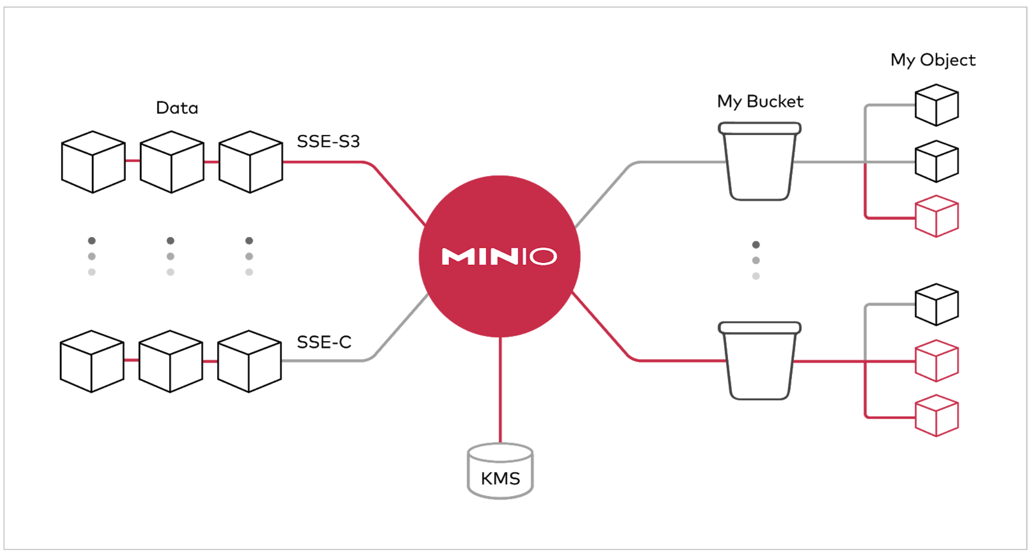


Figure 7: Encryption and WORM protect data in flights and at rest..

Global Federation

The modern enterprise has data everywhere. MinIO allows those various instances to be combined to form a unified global namespace. Any number of MinIO servers can be combined into a Distributed Mode set and multiple Distributed Mode sets can be combined into a MinIO Server Federation. Each MinIO Server Federation provides a unified admin and namespace.

A MinIO Federation Server supports an unlimited number of Distributed Mode sets.

The impact of this approach is that an object store can scale massively for large, geographically distributed enterprise while retaining the ability to accommodate a variety of analytical approaches (S3 Select, MinSQL, Spark, Hive, Presto, TensorFlow, H20) from a single console.

There are multiple benefits to MinIO's cluster and federation architecture:

- Each node is an equal member of a MinIO cluster. There is no master node.
- Each node can serve requests for any object in the cluster, even concurrently.
- Each cluster uses a Distributed Locking Manager (DLM) to manage updates and deletes to objects.
- The performance of an individual cluster remains constant as you add more clusters to the federation.
- Failure domains are kept within the cluster. An issue with one cluster does not affect the entire federation.

When deploying a cluster, it is recommended that you use a programmable domain name service (DNS), such as coreDNS, to route HTTP(S) requests to the appropriate cluster. Also, use a load balancer to balance the load across the servers in a cluster. Global configuration parameters can be stored and managed in etcd (an open-source distributed key-value store).

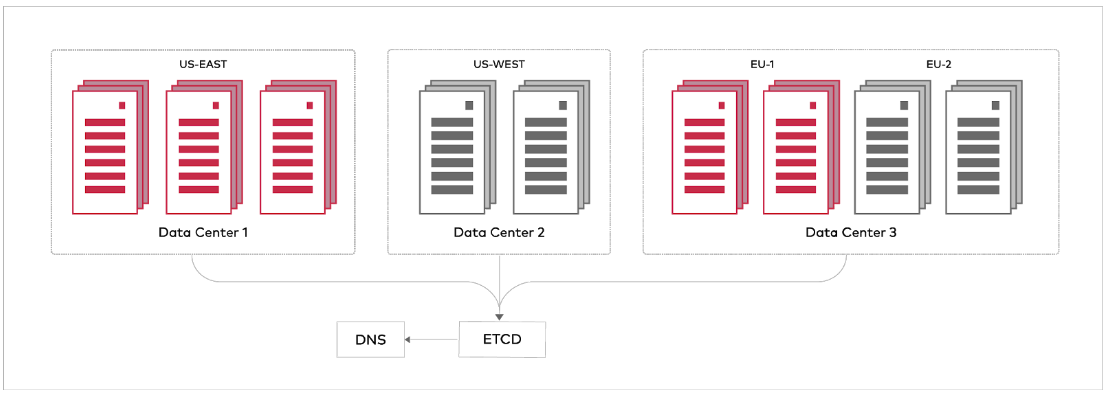


Figure 8: Global federation enables almost infinite scalability..

Multi-Cloud Gateway

All enterprises are adopting a multi-cloud strategy.

To support hybrid cloud initiatives, MinIO can be deployed in gateway mode to leverage public cloud resources. Leveraging the same binary, MinIO enables companies to run their applications on premises or in the public cloud with no modification. This minimizes operational overhead, and provides flexibility to move data and applications as business requirements change, not locking into a specific cloud provider or proprietary architecture. To achieve this requires that your bare-metal virtualization containers and public cloud services (including non-S3 providers like Google, Microsoft and Alibaba) look identical. MinIO runs on bare metal, network attached storage and every public cloud. More importantly, MinIO ensures your view of that data looks exactly the same from an application and management perspective via the Amazon S3 API.

MinIO, can go even further, making your existing storage infrastructure compatible with Amazon S3. The implications are profound. Now organizations can truly unify their data infrastructure - from file to block, all appearing as objects accessible via the Amazon S3 API without the requirement for migration.

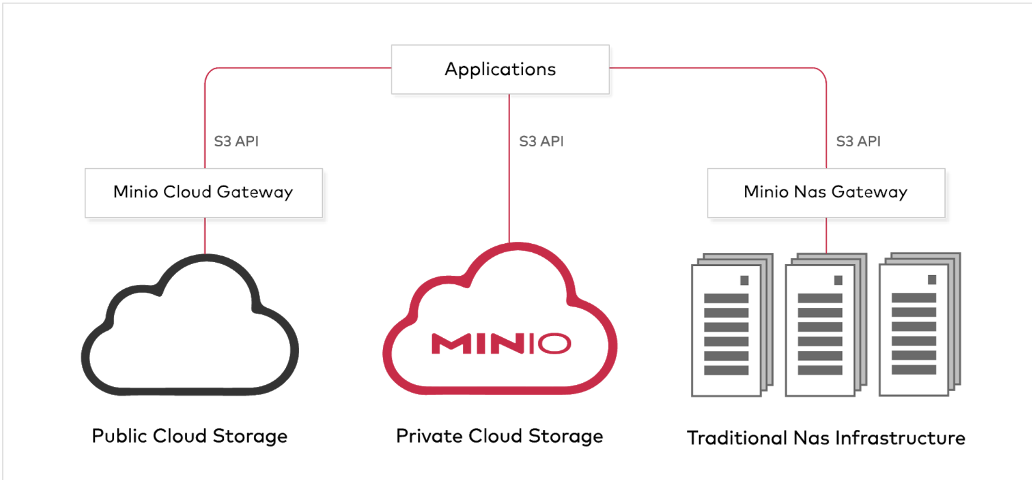
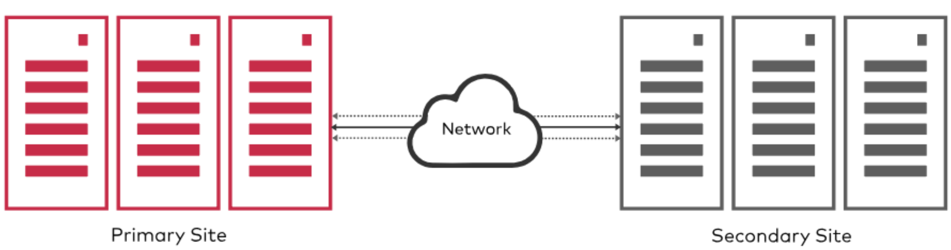


Figure 9: Gateway mode is designed to make every cloud and NAS look like S3.

Continuous Replication

The challenge with traditional replication approaches is that they do not scale effectively beyond a few hundred TB. Having said that, everyone needs a replication strategy to support disaster recovery (DR) and that strategy needs to span geographies, data centers and clouds. MinIO's continuous replication is designed for large scale, cross data center deployments. By leveraging Lambda compute notifications and object metadata it can compute the delta efficiently and quickly.

Lamba notifications ensure that changes are propagated immediately as opposed to traditional batch methods. Continuous replication means that data loss will be kept to a bare minimum should a failure occur - even in the face of highly dynamic datasets. Finally, like all that MinIO does, continuous replication is multi-vendor, meaning that your backup location can be anything from NAS to the public cloud.




```
mc mirror --watch --recursive dc1/ dc2/
```

Figure 10: MinIO's continuous replication approach safeguards even dynamic data.

Metadata Architecture

MinIO has no separate metadata store. All operations are performed atomically at object level granularity. This approach isolates any failures to be contained within an object and prevents spillover to larger system failures. Each object is strongly protected with erasure code and bitrot hash. You can crash a cluster in the middle of a busy workload and still not lose any data. Another advantage of this design is strict consistency which is important for distributed machine learning and big data workloads.

Cloud Native

The multi-instance, multi-tenant design of MinIO enables Kubernetes-like orchestration platforms to seamlessly manage storage resources just like compute resources. Each instance of MinIO is provisioned on demand through self-service registration. Traditional storage systems are monolithic and compete with Kubernetes resource management. MinIO is lightweight and container friendly so you can pack many tenants simultaneously on the same shared infrastructure.

Lambda Function Support

MinIO supports Amazon compatible Lambda event notifications which enables applications to be notified of individual object actions such as access, creation, and deletion. The events can be delivered using industry standard messaging platforms like Kafka, NATS, AMQP, MQTT, Webhooks, or a database like Elasticsearch, Redis, Postgres, and MySQL.

Benchmark Performance: S3 Bench

Performance claims require context and benchmarks. MinIO tests against a number of different benchmarks from S3 to DFS.io and TPC. The following represents the summary results of our S3 Bench testing on commodity and high performance hardware. Full documentation of the testing, setup and environments can be found on MinIO's website.

Our HDD results running on 16 node Minio cluster were:

Setup	Avg Read Throughput (GET)	Avg Write Throughput (PUT)
Distributed	10.81 GB/s	8.57 GB/s
Distributed with Encryption	9.38 GB/s	6.91 GB/s

Our NVMe results running on an 32 node MinIO cluster were:

Setup	Avg Read Throughput (GET)	Avg Write Throughput (PUT)
Distributed	183.2 GB/s	171.3 GB/s
Distributed with Encryption	162 GB/s	114.7 GB/s

An Enduring Commitment to Open Source

MinIO operates under the Apache V2 license. The company's products are 100% open source.

The advantages of open source are well documented. These include the avoidance of vendor lockin, security, consistent innovation, transparency, and the reliability that comes with millions of community members hammering every release from every possible angle.

MinIO remains the owner of the MinIO object storage project and as such controls the quality and development through its weekly release cadence. MinIO runs a suite of acceptance tests for every pull request and every MinIO server release.

Understanding the SUBNET Subscription Offering

While MinIO is available under the open source Apache V2 license, many customers choose to purchase the software on an annual subscription basis. Their reasons for doing so differ, but they are unified in the value they see in the software coupled with a desire to have a deeper relationship with the team behind MinIO.

The SUBNET subscription includes the object storage suite and all new features, individually tracked and prioritized security updates and bug fixes, advanced diagnostics, real time engineering support, customized release management and support for older, production implementation of MinIO.

SUBNET provides an extra measure of assurance to our enterprise customers with production deployments ranging from Terabyte to Exabyte.

Conclusion

MinIO is the fastest growing object storage system in the world for a reason. It was designed from scratch to be a key part of the modern data stack solving critical problems for enterprises while seamlessly integrating with its existing data and application infrastructure. It delivers performance, scalability and simplicity alongside an enterprise grade feature set.

More than 200 enterprises across the healthcare spectrum use MinIO today, to power their products or for high performance object storage use cases. That list includes IQVIA, Unitedhealthcare, Cerner Corporation, McKesson, Stanford Hospital and Clinics, Kaiser Permanente, The Methodist Hospital, Epic Systems Corporation, National Institutes of Health,Fairview Health Services, BCBS of Oregon, North Carolina, Iowa, Illinois, Anthem, University Health Network and the Cleveland Clinic.

Reach out to us at hello@min.io to learn more about how we can support your specific storage requirements.