Executive Summary

MinIO is a high performance, distributed object storage system. By following the methods and
design philosophy of hyperscale computing providers, MinIO delivers superior performance,
resilience and 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 delivering massive primary
storage across a range of use cases from Kubernetes-powered cloud applications to
AI/ML/advanced analytics 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 is a pioneer in the development of cloud-native object storage, refining and perfecting
many of the features, protocols and APIs that have come to define best in class. This is evidenced
by the more than 400M Docker pulls, 23K+ GitHub stars and the thousands of production
deployments across every continent.

This paper details the philosophical approach and technical attributes of MinIO and why those
attributes are important to any enterprise seeking to develop or migrate to an object storage
centric, microservices architecture across the public and private cloud.
The Enterprise Challenge

How enterprises store, access, move and analyze data is undergoing massive change. Driven by the storage and compute efficiencies made possible by disaggregation, enterprises are finding that their investments in traditional storage solutions like Hadoop HDFS are now obsolete. The weapon of elite hyperscalers, disaggregation offers multiple benefits, but the two largest are economics and performance oriented use cases like machine learning and advanced analytics. As a result, enterprises are rearchitecting their data infrastructures to take advantage of this separation.

The reasons are straightforward. File and block protocols are complex, have legacy architectures that impede innovation, are limited in their ability to scale or are compromised from a performance perspective. Examples of these limitations include the aforementioned aggregation of compute and storage but also include replication, security, encryption and data mobility.

The winner in this transformation is cloud-native, object storage.

Storage as a Service or STaaS is the second-fastest growing cloud workload worldwide, representing a USD 4.8 billion annual market. Data is growing exponentially every year and by 2025, experts predict that the world will create and replicate 163 zettabytes (ZB) of data. The vast majority of that will be unstructured or semi-structured.

Fueling that growth is a focus on big data applications, Internet of Things (IoT) and artificial intelligence (AI) workloads. These workloads demand high rates of throughput, excellent data integrity, and a cost-effective deployment model.
Simple, powerful and with unlimited scalability, modern object storage has moved out of backup and into the application and analytic workflow. A reduced set of storage APIs, accessed over HTTP RESTful services mean that these cloud-native solutions are lightweight enough to be packaged with the application stack.

Formal diagram: "Modern Cloud-Native Workloads" vs "Traditional IT Workload" categorized as "Slow" or "Fast" with "Modern Object Storage" vs "Block Storage" and "File Storage".

---

**The Philosophy Of The Cloud**

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 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 and 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 architected to deliver performance, scale and resiliency. As a software-defined solution, MinIO can be paired with hundreds of different compute and storage configurations from Intel Cascade Lake, ARM Graviton, or Atom processors on the compute side to Optane and NVMe SSDs and traditional spinning disk.

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 inline erasure code, bitrot protection, immutability, active-active replication and other features.
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. Further, extensive scripting capabilities enable automation for DevOps teams.

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 Go, 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 high-performance access to big data, analytic and machine learning workflows require server-side filtering features - also referred to as “predicate pushdown”.

MinIO has developed a SIMD accelerated version 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 the 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. This capability works for objects in CSV, JSON and Parquet formats and is effective on compressed objects as well.

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 data and parity blocks - although these can be configured to any desired redundancy level. This means that in a 12 drive setup with 6 parity configuration, an object is striped 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 limitations of 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. Replication results in 3 or more copies of the object on each of the sites. Erasure-code offers a significantly higher level of protection while only consuming a fraction of the storage space as compared to replication.

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

BitRot Protection

Silent data corruption, or bitrot is a serious problem for drives resulting in the corruption of data without the user’s knowledge. As the drives get larger and larger and the data needs to persist for many years, this problem is more common than we imagine. The data bits decay when the magnetic orientation flips and loses polarity. Even solid state drives are prone to this decay when the electrons leak due to insulation defects. There are also other reasons such as wear and tear, voltage spikes, firmware bugs and even cosmic rays.

MinIO’s SIMD accelerated implementation of the HighwayHash algorithm ensures that it will never return corrupted data - it captures and heals corrupted objects on the fly. Integrity is ensured from end to end by computing hash on WRITE and verifying it on every READ 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 per core on Intel CPUs.
Identity and Access Management

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

![Identity Protection and Single Sign-On (SSO) are critical enterprise features.](image)

**Encryption**

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 ensures 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 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.

![Encryption and WORM protect data in flights and at rest.](image)
Finally, MinIO has introduced its own Key Encryption Service (KES). Stateless and distributed (KES) was designed to be run inside Kubernetes and distribute cryptographic keys to performance oriented applications. KES operates as a bridge between a central KMS and cloud-native applications, as an abstraction layer over different KMS vendors and as a scale-out load balancer for cryptographic operations in distributed systems.

**Lifecycle Management + Immutability**

MinIO supports object locking, versioning, legal holds and governance/retention modes for objects/buckets. This capability is critical for ransomware use cases and can be used in conjunction with leading backup vendors to ensure fast backup/restore across multiple workloads. MinIO’s implementation earned validation from Cohasset Partners that MinIO meets the requirements of SEC 17a-4(f), FINRA 4511(c) and CFTC 1.31(c)-(d).

**Scalability and Distribution**

MinIO is a multi-tenant and multi-user system and is designed to scale seamlessly from TBs to any size. The tenants are fully isolated from each other with their own instances of MinIO clusters. Each tenant in turn may have multiple users with varying levels of access privileges. Each tenant cluster operates independently of each other. Each cluster is a collection of fully symmetric and distributed servers sets that participate equally in serving the objects. Standard HTTP load balancers or round-robin DNS may be employed. A single cluster may span an entire data center and grow to 100s of petabytes.

Within a cluster, racks of homogenous servers are grouped into zones. Zones are the basic unit of expansion and they bring the concept of rack-awareness and failure-domains. A zone can be as small as four servers and as large as multiple racks. A cluster is scaled by adding one or more zones at a time. There is no rebalancing penalty for scaling. Zones also allow heterogeneous expansion of the cluster.

Inside each zone is a collection of distributed erasure-sets. Each erasure-set contains up to a maximum of 16 drives. There is no fixed limit to the number of erasure-sets within a zone or the number of zones. Objects are striped across all the drives within an erasure-set with erasure-code and bitrot protection. An erasure-set is the fundamental unit of data protection and high-availability within the data-center. Every operation in MinIO is atomic, transactional and strictly consistent. A distributed quorum lock is acquired only at the time of namespace commit at an object-level granularity. The entire cluster is designed to be heavily concurrent and resilient.

When a new object enters the system, the endpoint URL and the bucket DNS name determines the physical location of the cluster. Within a cluster, a zone with the maximum amount of free drive space is chosen and further within it, an erasure-set is chosen using the deterministic hashing algorithm. This architecture allows applications to scale geographically using the most proven practices used by the hyper-scalers.
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 such as NAS and Hadoop HDFS 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.
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 feature uses the lambda notification API to track changes across Petabytes of data. This approach pushes changes instantly to the remote sites without requiring expensive namespace scans and batched operations. When both the sites run MinIO servers, you may enable the active-active server-side replication functionality. Client-side replication is recommended for replicating data between MinIO and third-party object storage or NAS vendors.

![Diagram of MinIO's continuous replication approach](image)

Metadata Architecture

MinIO has no separate metadata store. All operations are performed atomically at object level granularity with strong consistency. 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.

Kubernetes Native

The multi-instance, multi-tenant design of MinIO enables orchestration platforms like Kubernetes to seamlessly manage storage resources just like compute resources. Each instance of MinIO is provisioned on demand through self-service registration. Travel 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 serverless 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 such as Elasticsearch, Redis, Postgres, and MySQL.
**Sidekick**

Given MinIO’s architecture, standard HTTP load balancer or round-robin DNS may be employed. For high performance applications, however, there may be a need for a more streamlined approach. Traditional load balancer appliances have limited aggregate bandwidth and introduce an extra network hop. This architectural limitation is also true for software-defined load balancers running on commodity servers.

Sidekick solves the network bottleneck challenge by taking a sidecar approach instead. It runs as a tiny sidecar process alongside of each of the client applications. This way, the applications can communicate directly with the servers without an extra physical hop. Since each of the clients run their own Sidekick in a share-nothing model, it can be scaled to any number of clients.

![Diagram of Sidekick architecture](image)

In a cloud-native environment like Kubernetes, Sidekick runs as a sidecar container. Sidekick can be added to existing applications without any modification to the application binary or container image.

**Benchmark Performance**

MinIO’s performance claims are backed by well-documented benchmarks. MinIO tests against a number of different benchmarks from Warp, S3-benchmark to Hadoop DFSIO. 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 results running on a 8 node MinIO cluster can be summarized as follows:

<table>
<thead>
<tr>
<th>Setup</th>
<th>Avg Read Throughput (GET)</th>
<th>Avg Write Throughput (PUT)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distributed</td>
<td>46.54 GB/s</td>
<td>34.4 GB/s</td>
</tr>
<tr>
<td>Distributed with Encryption</td>
<td>46.4 GB/s</td>
<td>34.6 GB/s</td>
</tr>
</tbody>
</table>

Our results running on a 32 node MinIO cluster can be summarized as follows:

<table>
<thead>
<tr>
<th>Setup</th>
<th>Avg Read Throughput (GET)</th>
<th>Avg Write Throughput (PUT)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distributed</td>
<td>183.2 GB/s</td>
<td>171.3 GB/s</td>
</tr>
<tr>
<td>Distributed with Encryption</td>
<td>162 GB/s</td>
<td>114.7 GB/s</td>
</tr>
</tbody>
</table>

In addition, MinIO has compared itself to HDFS across key tests with outstanding results, proving that enterprises do not need to trade off performance to achieve the benefits of disaggregation.

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>HDFS</th>
<th>MinIO</th>
<th>MinIO is X% faster</th>
</tr>
</thead>
<tbody>
<tr>
<td>Terasort</td>
<td>1005s</td>
<td>820s</td>
<td>22.5%</td>
</tr>
<tr>
<td>Sort</td>
<td>1573s</td>
<td>793s</td>
<td>98.3%</td>
</tr>
<tr>
<td>Wordcount</td>
<td>1100s</td>
<td>787s</td>
<td>39.7%</td>
</tr>
</tbody>
</table>

**An Enduring Commitment to Open Source**

MinIO products are 100% open source under the Apache License v2 and the GNU Affero General Public License, Version 3.0 (AGPLv3). It is MinIO’s commitment that as long as it is independent that it will continue to be 100% open source.

The advantages of Open Source are well understood. 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 MinIO Subscription Network

While MinIO is available under the FOSS licences, 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 MinIO Subscription Network benefits fit into three core categories:

- **Commercial License** - A commercial license provides exceptions to the obligations inherent in the MinIO GNU AGPL v3 license. As the copyright holder, only MinIO can provide this exception. While many enterprises require commercial licenses where AGPL v3 is present, subscribers still enjoy the benefits associated with open source - namely freedom from lock-in and freedom to inspect the source.

- **Data Loss Prevention** - Large scale data infrastructure must consider and plan for failure as part of the design. The MinIO Subscription Network delivers access to technology and talent to manage and minimize this risk. These include direct-to-engineering support, one hour SLA, access to the Panic Button, performance and optimization audits as well as key diagnostic capabilities designed to assess the health of customer deployments.

- **Data Breach Prevention** - Modern data infrastructure is in a state of constant change, and change brings risk. Customers of the MinIO Subscription Network have 24/7/365 access to cryptographic experts, quarterly security audits, and a suite of other technologies that will ensure MinIO’s industry leading security features are properly configured while proactively identifying vulnerabilities and evolving risk.

The MinIO Subscription Network offers two tiers - Standard and Enterprise, which customers select based upon their SLA and legal requirements. The Standard Tier is priced at $.01 per GB per month. The Enterprise Tier is priced at $.02 per GB per month. Capacity minimums start at 25 TB and 100 TB respectively. The ceiling function turns off the capacity meter at 5PB or 10PB depending on the plan. No further charges above that threshold are applied. Pricing is month to month and subscribers can cancel at any time and with no questions asked.

**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.

As importantly, MinIO is 100% open source with all of the attendant benefits. Finally, for our production customers we offer the security that comes from a direct engagement with MinIO engineering via the SUBNET subscription offering on a subscription basis.

The result is the industry’s most comprehensive solution for the rapidly growing world of object storage.
About MinIO

Founded in 2014, MinIO is now the world’s fastest growing object storage system. Backed by some of the smartest minds in storage and venture capital including Nexus, General Catalyst, Dell Technologies Capital, Intel Capital, AME Cloud Ventures and key angel investors, the company has raised $23.3M through its Series A round.

Additional Information
Email: hello@Min.io
MinIO Inc.
530B University Avenue,
Palo Alto, CA 94301

Resources
https://min.io
https://docs.min.io/
https://blog.min.io/

© 2020 MinIO, Inc. *numbers in map as of October 2020