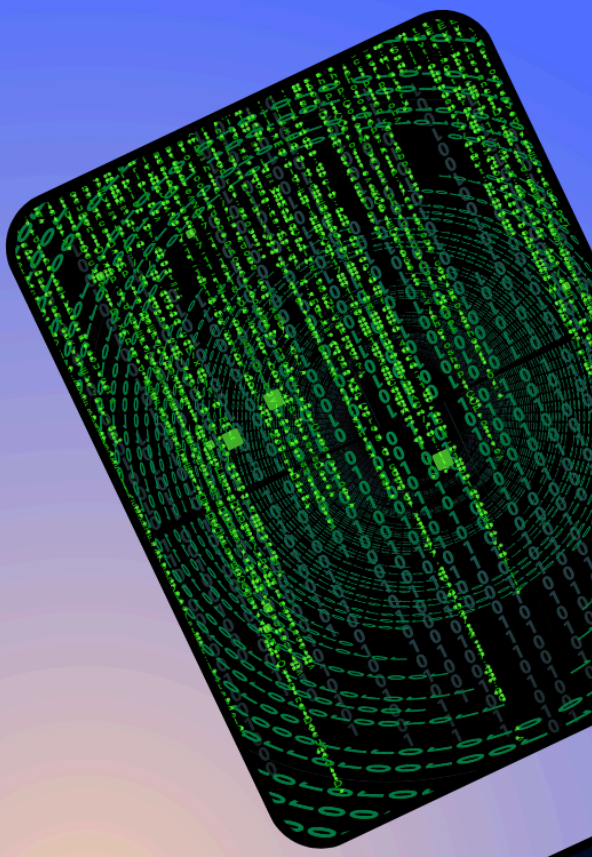
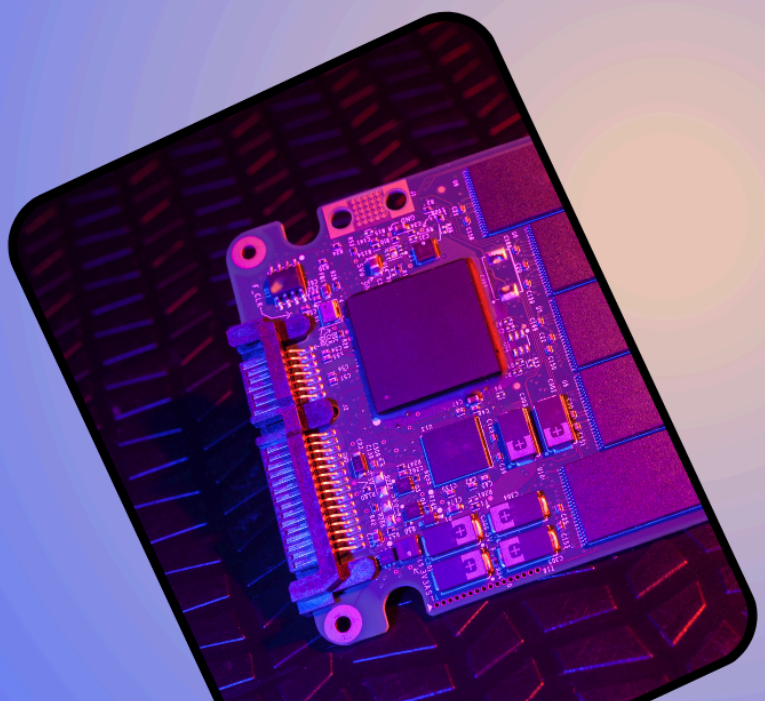
simplyblock

HIGH-PERFORMANCE BLOCK STORAGE FOR aws

CHEAPER and FASTER ALTERNATIVE to Amazon EBS®





Whitepaper

PROBLEM

Cloud computing users, without a doubt, tremendously benefit from the *ease-of-operations* and *-management*, the *on-demand availability*, the *flexible scaling of resources* and the general *reliability* of cloud infrastructure services, as well as the ability to combine them with a multitude of fully managed and operated, yet powerful platform services.

At the same time, they suffer from drawbacks of cloud services when compared to self-owned or rented hardware-based data center systems. One of these drawbacks is related to high-performance data storage solutions. This is especially true, when running io-intensive services, such as databases, in hosted Kubernetes environments.

AWS offers Amazon EBS (*Elastic Block Storage*) as their native high-performance block storage solution for the AWS cloud. This storage is available for all types of EC2 instances, but also for other products such as Amazon EKS (AWS' managed Kubernetes). Amazon EBS is a mature product, but lacks in comparison to the last-generation data center storage solutions.

Shortcomings exist in many different areas, technical ones, as well as operational, or business related issues:

- The **storage cost per GB is high**
- The **cost per performance unit (IOPS) is particularly high**
- **Amazon EBS is an allocation-based, not a usage based pricing model**, meaning, you have to allocate the necessary disk space upfront, and pay based on that allocation. With your own hardware, you can thinly provision your volumes and "pay" for what you use only.
- **Cross-site replication of block storage works only within AWS**. While there are other replication solutions, they need to be deployed separately and aren't block level based, hence, more complicated or restricted.
- **The durability of Amazon EBS volumes** (gp2, gp3, io1) **is only at 99.9%**. To provide **sufficient protection against any production data from loss** multiple volumes need to be combined (software raid), and further volumes may be required for fail-over requirements, increasing the price.

While customers can choose from hundreds of AWS services, storage for data of io-heavy systems can quickly build up to 50% (or more) of their cloud bill on AWS, with the biggest chunk being related to Amazon EBS (on average more than 20%). We have estimated that AWS earns more than 4 billion USD annually just with their EBS service.

SOLUTION

Simplyblock offers an alternative to Amazon EBS on Kubernetes services, such as Amazon EKS. Utilizing our Vash™ cluster technology, we enable significant cost reductions, while providing higher IOPS values. Simplyblock is **50-80% cheaper than a similar Amazon EBS volume**, delivers **lower and more predictable latency** (thanks to NVMe® over Fabric), and combines that with the **highest possible reliability** (both durability and availability) of up to **99.9999%**. All of this without sacrificing beloved on-prem SAN features such as **resource pooling through thin provisioning**, **encryption**, **multipathing**, and others.

Simplyblock's storage cluster solution is deployed into your AWS account and runs on EC2 instances with local NVMe storage attached. Our automated installation routines help setting up the necessary cluster services in minutes, not hours.

The cluster runs nearly operations-free: with automated **health checks,** provides **self-healing capabilities** and autonomously **recovers** from device or node failures **in the background without any impact on ongoing service**.

We top it off with **zero-downtime scalability**. It is also possible to **grow or shrink the cluster at any time** by removing nodes or adding new nodes, **without service interruption**. The cluster automation **automatically migrates data between disks and nodes** to balance the load and capacity utilization across the cluster.

Access to the storage cluster is provided through our CSI (Container Storage Interface) driver, and is available as a Kubernetes StorageClass, making it a drop-in replacement for any other, already existing, StorageClass, by just exchanging the name.

Storage consumers can reside in any AWS account, as long as it is located in the same availability zone.

The software is **multi-tenant-capable**, thus providing a simple solution for Managed Service Providers to enable sharing storage resources with multiple customers, without sacrificing isolation or introducing privacy concerns.

COMPARISON (AMAZON EBS vs SIMPLYBLOCK)

Amazon EBS is expensive, especially when data-intensive workloads require a high amount of IOPS (IO operations per second) and storage capacity. As mentioned above, storage cost can easily reach 50% of the cloud spend. That said, people are looking at storage cost head first when requested to reduce the overall cloud cost.

Simplyblock operation cost, when compared to Amazon EBS volumes, commonly provides over 60% savings, while offering better performance (predictable low latency, higher IOPS).

Amazon EBS vs simplyblock	**io1**	**io2**	**io2 block express***	simplyblock
Monthly Cost per 1K IOPS-Provisioned	$65	$65	$65	$7
Monthly Cost per TB-Provisioned	$125	$125	$125	$65
Durability / Availability	99.9%	99.9999%	99.9999%	99.9999%
Average Access Latency	Single Digit Milliseconds	Single Digit Milliseconds	Sub-Millisecond	Sub-Millisecond*
Maximum IOPS per Volume	64,000	64,000	256,000	256,000
Maximum IOPS per GB	50	500	1,000	1,000
Maximum Throughput per Volume	1 GB/s	1 GB/s	4 GB/s	4 GB/s

* availability limited to some instance types
** based on i3en.6xlarge, 3-year reservation, region EU East, assumption: average ratio of provision-to-utilization is 1.5x
*** minimum is about 20 microseconds in same availability zone and placement group

ARCHITECTURE

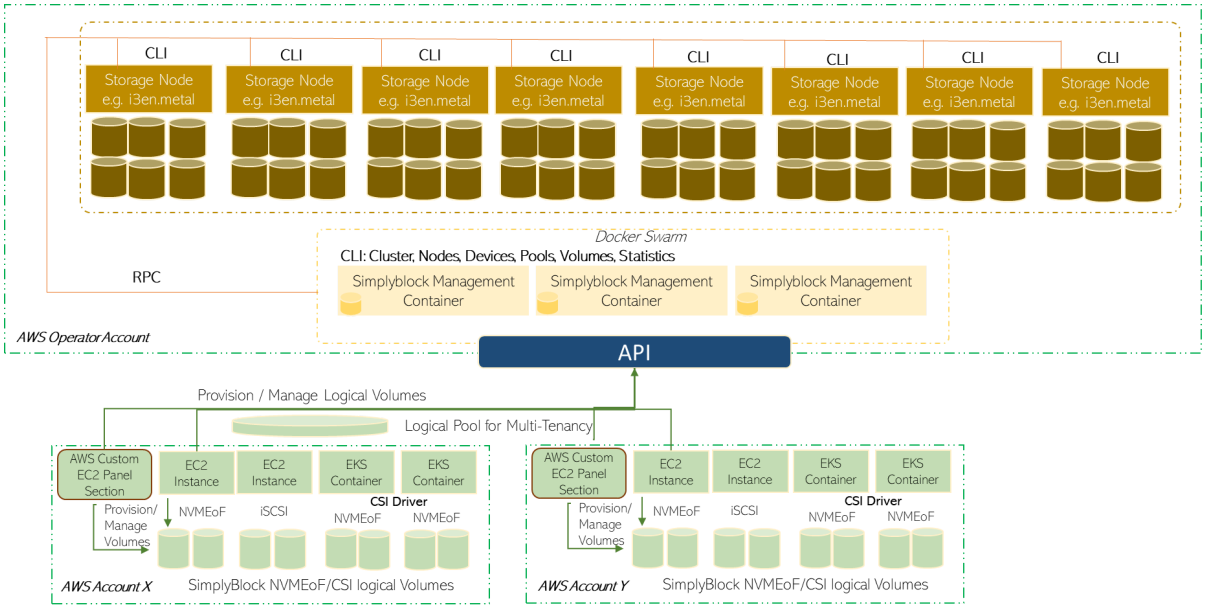
A **simplyblock storage cluster** consists of management nodes and storage nodes. Storage nodes must have a gp3, io1, or io2 NVMe® backed volume attached. The best ratio between performance and costs is currently achieved with instances of type i3en (for storage nodes) and m6i (for management nodes).

The minimum configuration contains one storage node, but we recommend a minimum of 4 nodes for high-availability and data endurance (99.9999%). For optimal cost to capacity efficiency we **recommend clusters with 8 or more nodes.** There is no upper limitation in simplyblock. Nodes can be added at any time without service impact.

Management nodes host the management CLI, the external API, and a fully replicated, local key-value store for the cluster configuration.

Storage nodes run the simplyblock storage driver stack within spdk and a local controller component, as well as a CLI.

Volumes can be provisioned from within any AWS account, within the same availability zone as the storage cluster, using the API or directly from Kubernetes deployment descriptors through a Persistent Volume Claim (PVC), using the provided StorageClass.



COST SAVINGS EXPLAINED

Cost Savings come from multiple factors. The biggest one is the **difference between costs of IOPS**. As a cluster consists of instances with local NVMe storage, meaning, there is **no extra charge for IOPS**.

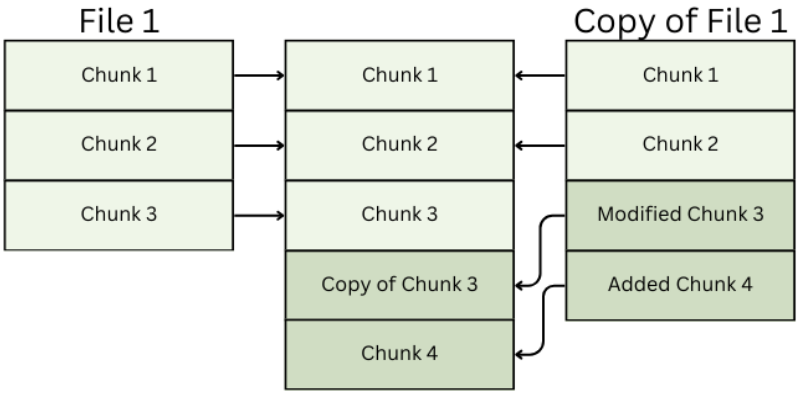
The largest Amazon EC2 instances can deliver up to about 3 million IOPS, **creating a cluster of 8 nodes with up to 24 million IOPS**. Provisioning a similar Amazon EBS volume with io2, would cost **1,536,000 USD per month.** At the same time the **cluster cost is about 59,500 USD per month** (EAST-1) provisioned on-demand. With a **three year reservation the cost can be dropped down to about 23,800 USD**.

Amazon EBS offers a free baseline of 3,000 IOPS with gp3 volumes. That means the cheapest native option is rated at about 80 USD per GB and month. Distributing this volume across a simplyblock cluster, we can create volumes with 25 USD per GB and month. At the same time, simplyblock provides a much higher IOPS value (15,000 - 30,000).
There are four main reasons for the decreased price, while enabling higher IOPS.

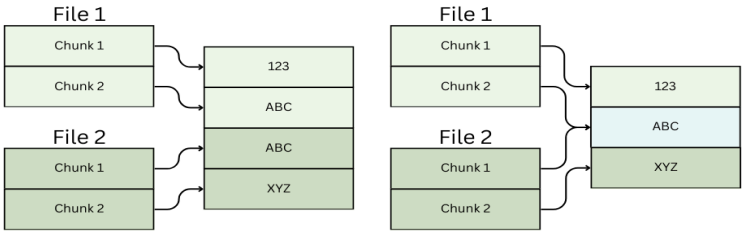
1. **Instance reservation**, not available for Amazon EBS storage volumes, is heavily utilized by simplyblock, and enables us to substitute expensive large volumes for a multitude of conflated (via storage distribution) small ones. A 3-year reservation, as an example, will decrease the cluster cost by about 60%.
2. **Copy-on-write (COW)** on logical volumes enables zero-cost copies. That means, **volume clones are free (cost and space usage)**. A copy-on-write storage only stores changed (or mutated) data. You can have **hundreds of copies** referring to the same unchanged file **at no extra cost**. That said, if all containers contain the same base images, you only use storage capacity for the files specific to a container, not the shared basis.
3. **Compression and Deduplication (available in 2024)** are built-in, achieving **compaction results of up to 3:1**, meaning a **data reduction by up to 67%** or a stored volume of 3 TB will be effectively 1 TB of raw storage (example of 3 PostgreSQL nodes, 1 primary, 2 read-replicas).
4. **Thin provisioning** enables **significantly improved resource utilization** due to **overcommitting storage capacity** and **storage pooling**. With thin provisioning, large volumes will only use as much data as is currently stored within them, while overcommitment enables a higher overall committed storage volume (across all logical volumes) than available in the cluster. The combination of overcommitment and thin provisioning decreases the storage cost to a bare minimum at any given time, without limiting future growth.

Some of the above features may need a bit more explanation, specifically **copy-on-write**, **deduplication**, **thin provisioning**, and **overcommitment**.

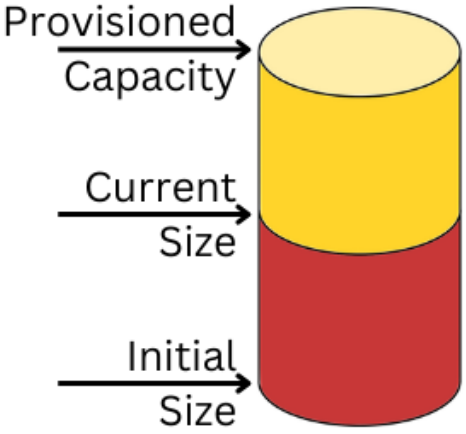
Copy-on-write or COW is a common technique to shrink the overall usage of memory (like RAM or disk storage) consumption. It assumes that copies modify very limited regions of the shared resource, such as a file. In most cases it will not modify it at all. Therefore, a single physical copy of the file can be stored, no matter how many virtual file copies refer to it. The moment one of the virtual files wants to change a part of the file, a part of the file or the complete file is physically copied and the virtual file copy changed to refer to that new physical copy.



Changing files will eventually create multiple physical copies, however, the content of the copies may end up being identical (overall or in parts). This is where deduplication comes in. Deduplication looks around the storage and finds identical blocks of data, and deduplicates them, meaning, to move all referring handles to just one copy of the data and marking the remaining copies as free available space.

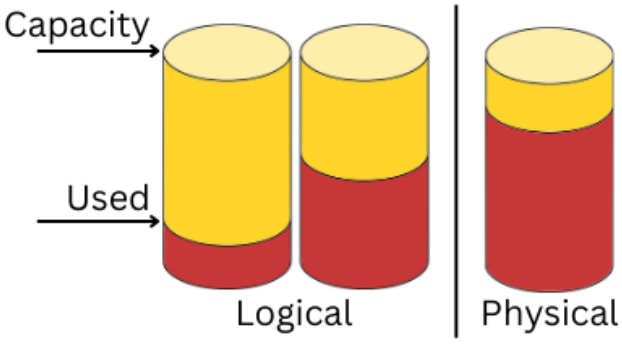


While copy-on-write and deduplication are ongoing operations, thin provisioning and overcommitment optimize initial resource sharing. Logical volume utilization fluctuates between a minimum and a maximum. Most often, volume usage starts small and grows over time. We can make use of this behavior and provision a logical volume with a theoretical size of 3 TB. Since it's initially empty, it will effectively utilize no space at all though. Storing data into the logical volume will have the effective size grow with it. The concept is very similar to sparse files.



It also enables us to provision a much higher amount of storage capacity than actually available at the time of provisioning. Imagine a physical storage capacity of 20 TB, and 20 logical volumes, each thinly provisioned with a maximum capacity of 5 TB.

In the physical world we would need 100 TB (20 x 5 TB) to fulfill this request. However, since we don't expect the logical volumes to grow quickly we can overcommit at the time of creation, as long as we make sure to increase the physical capacity before we run out of memory. This concept is similar to how overcommitment of CPU and RAM works for virtual machines.



Within a larger storage cluster, the general fluctuation is much smaller due to statistical averaging of logical volume usage (some go up while others go down). This means, the physical storage amount can be much smaller than the combined sizes of all the individual logical volumes. This leads to significant savings.

ABOUT US

Simplyblock is the first company to build a container storage system for Kubernetes, specifically designed for latency-sensitive, IO-intensive, cloud-native stateful workloads, such as databases, analytics, crypto / blockchain, document storages, and others.

A fully distributed storage solution, built upon our VASH™ technology, offering seamless scalability without downtimes, and is highly optimized for AWS NVMe® (io1/2) and gp3 volumes.

Simplyblock GmbH is based in Teltow, Brandenburg, Germany.