Amazon EBS FAQs

### 

### **General**

Q: Are Amazon EBS volume and snapshot ID lengths changing in 2018?

Yes, please visit [the EC2 FAQ](https://aws.amazon.com/ec2/faqs/#longer-ids) page for more details.

Q: What happens to my data when an Amazon EC2 instance terminates?

Unlike the data stored on a local instance store (which persists only as long as that instance is alive), data stored on an Amazon EBS volume can persist independently of the life of the instance. Therefore, we recommend that you use the local instance store only for temporary data. For data requiring a higher level of durability, we recommend using Amazon EBS volumes or backing up the data to Amazon S3. If you are using an Amazon EBS volume as a root partition, set the Delete on termination flag to "No" if you want your Amazon EBS volume to persist outside the life of the instance.

Q: What kind of performance can I expect from Amazon EBS volumes?

Amazon EBS provides four current generation volume types: Provisioned IOPS SSD (io1), General Purpose SSD (gp2), Throughput Optimized HDD (st1) and Cold HDD (sc1). These volume types differ in performance characteristics and price, allowing you to tailor your storage performance and cost to the needs of your applications. For more performance information see the [EBS product details page](https://aws.amazon.com/ebs/details/).

For more information about Amazon EBS performance guidelines, see [Increasing EBS Performance](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSPerformance.html).

Q: Which volume should I choose?

Amazon EBS includes two major categories of storage: SSD-backed storage for transactional workloads (performance depends primarily on IOPS) and HDD-backed storage for throughput workloads (performance depends primarily on throughput, measured in MB/s). SSD-backed volumes are designed for transactional, IOPS-intensive database workloads, boot volumes, and workloads that require high IOPS. SSD-backed volumes include Provisioned IOPS SSD (io1) and General Purpose SSD (gp2). HDD-backed volumes are designed for throughput-intensive and big-data workloads, large I/O sizes, and sequential I/O patterns. HDD-backed volumes include Throughput Optimized HDD (st1) and Cold HDD (sc1).

Q: How do I modify the capacity, performance, or type of an existing EBS volume?

Changing a volume configuration is easy. The [Elastic Volumes](https://aws.amazon.com/ebs/details/#elasticvolumes) feature allows you to increase capacity, tune performance, or change your volume type with a single CLI call, API call or a few console clicks. For more information about Elastic Volumes, see the [Elastic Volumes documentation](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-expand-volume.html).

Q: Are EBS Standard Volumes still available?

EBS Standard Volumes have been renamed to EBS Magnetic volumes. Any existing volumes will not have been changed as a result of this and there are no functional differences in the EBS Magnetic offering compared to EBS Standard. The name of this offering was changed to avoid confusion with our General Purpose SSD (gp2) volume type which is our recommended default volume type.

Q: Are Provisioned IOPS SSD (io1) volumes available for all Amazon EC2 instance types?

Yes, Provisioned IOPS SSD (io1) volumes are available for all [Amazon EC2 Instance Types](https://aws.amazon.com/ec2/instance-types/). To enable your EC2 instances to use the IOPS provisioned on an EBS volume consistently and predictably, you can launch selected EC2 instance types as EBS-optimized instances. [EBS-optimized instances](https://aws.amazon.com/ebs/details/#ebsoptimized) deliver dedicated throughput between Amazon EC2 and Amazon EBS, with options between 62.5 MB/s and 1,750 MB/s depending on the instance type used.

### **Performance**

Q: What level of performance consistency can I expect to see from my Provisioned IOPS SSD (io1) volumes?

When attached to EBS-optimized instances, Provisioned IOPS SSD (io1) volumes are designed to deliver within 10% of the provisioned IOPS performance 99.9% of the time in a given year. Your exact performance depends on your application’s I/O requirements.

Q: What level of performance latency can I expect to see from my Provisioned IOPS SSD (io1) volumes?

When attached to EBS-optimized instances, Provisioned IOPS volumes can achieve single digit millisecond latencies. Your exact performance depends on your application’s I/O requirements.

Q: Does the I/O size of my application reads and writes affect the rate of IOPS I get from my Provisioned IOPS SSD (io1) volumes?

Yes. For a given allocation of resources, the IOPS rate you get depends on the I/O size of your application reads and writes. Provisioned IOPS volumes process your application reads and writes in I/O sizes of 256KB or less. Every increase in I/O size above 256KB increases linearly the resources you need to achieve the same IOPS rate. For example, if you have provisioned a volume with 500 IOPS, that means that it can handle up to 500 256KB writes per second, 250 512KB writes per second, or 125 1024KB writes per second, and so on. You can use [Amazon CloudWatch](https://aws.amazon.com/cloudwatch/) to monitor your throughput and I/O sizes.

Q: What factors can affect the performance consistency I see with Provisioned IOPS SSD (io1) volumes?

Provisioned IOPS SSD (io1) volumes attached to EBS-optimized instances are designed to offer consistent performance, delivering within 10% of the provisioned IOPS performance 99.9% of the time over a given year. For maximum performance consistency with new volumes created from a snapshot, we recommend reading or writing to all of the blocks on your volume before placing it into service.

Another factor that can impact your performance is if your application isn’t sending enough I/O requests. This can be monitored by looking at your volume’s queue depth. The queue depth is the number of pending I/O requests from your application to your volume. For maximum consistency, a Provisioned IOPS volume must maintain an average queue depth (rounded to the nearest whole number) of one for every 500 provisioned IOPS in a minute. For example, for a volume provisioned with 1500 IOPS, the queue depth average must be 3. For more information about ensuring consistent performance of your volumes, see [Increasing EBS Performance](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSPerformance.html).

Q: What level of performance consistency can I expect to see from my HDD-backed volumes?

When attached to EBS-optimized instances, Throughput Optimized HDD (st1) and Cold HDD (sc1) volumes are designed to deliver within 10% of the expected throughput performance 99% of the time in a given year. Your exact performance depends on your application’s I/O requirements and the performance of your EC2 instance.

Q: Does the I/O size of my application reads and writes affect the rate of throughput I get from my HDD-backed volumes?

Yes. The throughput rate you get depends on the I/O size of your application reads and writes. HDD-backed volumes process reads and writes in I/O sizes of 1MB. Sequential I/Os are merged and processed as 1 MB units while each non-sequential I/O is processed as 1MB even if the actual I/O size is smaller. Thus, while a transactional workload with small, random IOs, such as a database, won't perform well on HDD-backed volumes, sequential I/Os and large I/O sizes will achieve the advertised performance of st1 and sc1 for a longer period of time.

Q: What factors can affect the performance consistency of my HDD-backed volumes?

Throughput Optimized HDD (st1) and Cold HDD (sc1) volumes attached to EBS-optimized instances are designed to offer consistent performance, delivering within 10% of the expected throughput performance 99% of the time in a given year. There are several factors that could affect the level of consistency you see. For example, the relative balance between random and sequential I/O operations on the volume can impact your performance. Too many random small I/O operations will quickly deplete your I/O credits and lower your performance down to the baseline rate. Your throughput rate may also be lower depending on the instance selected. Although st1 can drive throughput up to 500 MB/s, performance will be limited by the separate instance-level limit for EBS traffic. Another factor is taking a snapshot which will decrease expected write performance down to the baseline rate, until the snapshot completes. This is specific to st1 and sc1.

Your performance can also be impacted if your application isn’t sending enough I/O requests. This can be monitored by looking at your volume’s queue depth and I/O size. The queue depth is the number of pending I/O requests from your application to your volume. For maximum consistency, HDD-backed volumes must maintain an average queue depth (rounded to the nearest whole number) of four or more for every 1 MB sequential I/O. For more information about ensuring consistent performance of your volumes, see [Increasing EBS Performance.](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSPerformance.html)

Q: Can I stripe multiple volumes together to get better performance?

Yes. You can stripe multiple volumes together to achieve up to 75,000 IOPS or 1,750 MiB/s when attached to larger EC2 instances. However, performance for st1 and sc1 scales linearly with volume size so there may not be as much of a benefit to stripe these volumes together.

### **Snapshots**

Q: Will I be able to access my snapshots using the regular Amazon S3 API?

No, snapshots are only available through the Amazon EC2 API.

Q: Do volumes need to be un-mounted to take a snapshot?

No, snapshots can be done in real time while the volume is attached and in use. However, snapshots only capture data that has been written to your Amazon EBS volume, which might exclude any data that has been locally cached by your application or OS. To ensure consistent snapshots on volumes attached to an instance, we recommend detaching the volume cleanly, issuing the snapshot command, and then reattaching the volume. For Amazon EBS volumes that serve as root devices, we recommend shutting down the machine to take a clean snapshot.

Q: Does it take longer to snapshot an entire 16 TB volume as compared to an entire 1 TB volume?

By design, an EBS Snapshot of an entire 16 TB volume should take no longer than the time it takes to snapshot an entire 1 TB volume. However, the actual time taken to create a snapshot depends on several factors including the amount of data that has changed since the last snapshot of the EBS volume.

Q: Are snapshots versioned? Can I read an older snapshot to do a point-in-time recovery?

Each snapshot is given a unique identifier, and customers can create volumes based on any of their existing snapshots.

Q: How can I discover Amazon EBS snapshots that are shared with me?

You can find snapshots that are shared with you by selecting Private Snapshots from the list in the Snapshots section of the AWS Management Console. This section lists both snapshots that you own and snapshots that are shared with you.

Q: How can I find which Amazon EBS snapshots are shared globally?

You can find snapshots that are shared globally by selecting Public Snapshots from the list in the Snapshots section of the AWS Management Console.

Q: How can I find a list of Amazon public datasets stored in Amazon EBS Snapshots?

You can use the AWS Management Console to find public datasets stored as Amazon Snapshots. Log into the console, select the Amazon EC2 Service, select Snapshots and then filter on [Public Snapshots](https://console.aws.amazon.com/ec2/v2/home#Snapshots:visibility=public;sort=startTime). All information on public datasets is available in our [AWS Public Datasets](https://aws.amazon.com/public-datasets/) resource center.

### **Encryption**

Q: What is Amazon EBS encryption?

Amazon EBS encryption offers seamless encryption of EBS data volumes, boot volumes and snapshots, eliminating the need to build and maintain a secure key management infrastructure. EBS encryption enables data at rest security by encrypting your data using Amazon-managed keys, or keys you create and manage using the [AWS Key Management Service](https://aws.amazon.com/kms/) (KMS). The encryption occurs on the servers that host EC2 instances, providing encryption of data as it moves between EC2 instances and EBS storage. For more details, see Amazon EBS encryption in the [Amazon EC2 User Guide](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSEncryption.html).

Q: What is the AWS Key Management Service (KMS)?

AWS KMS is a managed service that makes it easy for you to create and control the encryption keys used to encrypt your data. AWS Key Management Service is integrated with other AWS services including Amazon EBS, Amazon S3, and Amazon Redshift, to make it simple to encrypt your data with encryption keys that you manage. AWS Key Management Service is also integrated with AWS CloudTrail to provide you with logs of all key usage to help meet your regulatory and compliance needs. To learn more about KMS, visit the [AWS Key Management Service](https://aws.amazon.com/kms/) product page.

Q: Why should I use EBS encryption?

You can use Amazon EBS encryption to meet security and encryption compliance requirements for data at rest encryption in [the cloud](https://aws.amazon.com/what-is-cloud-computing/). Pairing encryption with existing IAM access control policies improves your company’s defense-in-depth strategy.

Q: How are my Amazon EBS encryption keys managed?

Amazon EBS encryption handles key management for you. Each newly created volume gets a unique 256-bit AES key; Volumes created from the encrypted snapshots share the key. These keys are protected by our own key management infrastructure, which implements strong logical and physical security controls to prevent unauthorized access. Your data and associated keys are encrypted using the industry-standard AES-256 algorithm.

Q: Does EBS encryption support boot volumes?

Yes.

Q: Can I create an encrypted data volume at the time of instance launch?

Yes, using [customer master keys (CMKs)](https://docs.aws.amazon.com/kms/latest/developerguide/concepts.html#master_keys) that are either AWS-managed or customer-managed. You can specify the volume details and encryption through a [RunInstances API](https://docs.aws.amazon.com/AWSEC2/latest/APIReference/API_RunInstances.html) call with the [BlockDeviceMapping](https://docs.aws.amazon.com/AWSEC2/latest/APIReference/API_BlockDeviceMapping.html) parameter or through the Launch Wizard in the EC2 Console.

Q: Can I create additional encrpyted data volumes at the time of instance launch that are not part of the AMI?

Yes, you can create encrpyted data volume with either default or custom CMK encryption at the time of instances launch. You can specify the volume details and encryption through [BlockDeviceMapping](http://docs.aws.amazon.com/AWSEC2/latest/APIReference/API_BlockDeviceMapping.html) object in [RunInstances](http://docs.aws.amazon.com/AWSEC2/latest/APIReference/API_RunInstances.html) API call or through Launch Wizard in EC2 Console.

### **Billing and Metering**

Q: Will I be billed for the IOPS provisioned on a Provisioned IOPS volume when it is disconnected from an instance ?

Yes, you will be billed for the IOPS provisioned when it is disconnected from an instance. When a volume is detached, we recommend you consider creating a snapshot and deleting the volume to reduce costs. For more information, see the "Underutilized Amazon EBS Volumes" cost optimization check in [Trusted Advisor](http://aws.amazon.com/support/trustedadvisor/).  This item checks your Amazon Elastic Block Store (Amazon EBS) volume configurations and warns when volumes appear to be underused.

Q: Do your prices include taxes?

Except as otherwise noted, our prices are exclusive of applicable taxes and duties, including VAT and applicable sales tax. For customers with a Japanese billing address, use of AWS services is subject to Japanese Consumption Tax.

# Amazon EBS Volume Types

Amazon EBS provides the following volume types, which differ in performance characteristics and price, so that you can tailor your storage performance and cost to the needs of your applications. The volumes types fall into two categories:

* SSD-backed volumes optimized for transactional workloads involving frequent read/write operations with small I/O size, where the dominant performance attribute is IOPS
* HDD-backed volumes optimized for large streaming workloads where throughput (measured in MiB/s) is a better performance measure than IOPS

The following table describes the use cases and performance characteristics for each volume type:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Solid-State Drives (SSD)** | | **Hard disk Drives (HDD)** | |
| **Volume Type** | General Purpose SSD (gp2)\* | Provisioned IOPS SSD (io1) | Throughput Optimized HDD (st1) | Cold HDD (sc1) |
| **Description** | General purpose SSD volume that balances price and performance for a wide variety of workloads | Highest-performance SSD volume for mission-critical low-latency or high-throughput workloads | Low cost HDD volume designed for frequently accessed, throughput-intensive workloads | Lowest cost HDD volume designed for less frequently accessed workloads |
| **Use Cases** | * Recommended for most workloads * System boot volumes * Virtual desktops * Low-latency interactive apps * Development and test environments | * Critical business applications that require sustained IOPS performance, or more than 10,000 IOPS or 160 MiB/s of throughput per volume * Large database workloads, such as:   + MongoDB   + Cassandra   + Microsoft SQL Server   + MySQL   + PostgreSQL   + Oracle | * Streaming workloads requiring consistent, fast throughput at a low price * Big data * Data warehouses * Log processing * Cannot be a boot volume | * Throughput-oriented storage for large volumes of data that is infrequently accessed * Scenarios where the lowest storage cost is important * Cannot be a boot volume |
| **API Name** | gp2 | io1 | st1 | sc1 |
| **Volume Size** | 1 GiB - 16 TiB | 4 GiB - 16 TiB | 500 GiB - 16 TiB | 500 GiB - 16 TiB |
| **Max. IOPS\*\*/Volume** | 10,000 | 32,000\*\*\* | 500 | 250 |
| **Max. Throughput/Volume** | 160 MiB/s | 500 MiB/s† | 500 MiB/s | 250 MiB/s |
| **Max. IOPS/Instance** | 80,000 | 80,000 | 80,000 | 80,000 |
| **Max. Throughput/Instance**†† | 1,750 MiB/s | 1,750 MiB/s | 1,750 MiB/s | 1,750 MiB/s |
| **Dominant Performance Attribute** | IOPS | IOPS | MiB/s | MiB/s |

\* Default volume type for EBS volumes created from the console is gp2. Volumes created using the CreateVolume API default to either gp2 or standard according to region:

* standard: us-east-1, eu-west-1, eu-central-1, us-west-2, us-west-1, sa-east-1, ap-northeast-1, ap-northeast-2, ap-southeast-1, ap-southeast-2, ap-south-1, us-gov-west-1, cn-north-1
* gp2: All other regions

\*\* gp2/io1 based on 16 KiB I/O size, st1/sc1 based on 1 MiB I/O size

\*\*\* io1 volumes created in regions ap-northeast-3 and us-gov-west-1 are subject to a 20,000 IOPS limit.

† An io1 volume created before 12/6/2017 will not achieve this throughput until modified in some way. For more information, see [Modifying the Size, IOPS, or Type of an EBS Volume on Linux](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-modify-volume.html).

†† To achieve this throughput, you must have an instance that supports it. For more information, see [Amazon EBS–Optimized Instances](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSOptimized.html).

The following table describes previous-generation EBS volume types. If you need higher performance or performance consistency than previous-generation volumes can provide, we recommend that you consider using General Purpose SSD (gp2) or other current volume types. For more information, see [Previous Generation Volumes](https://aws.amazon.com/ebs/previous-generation/).

|  |  |
| --- | --- |
| **Previous Generation Volumes** | |
| **Volume Type** | EBS Magnetic |
| **Description** | Previous generation HDD |
| **Use Cases** | Workloads where data is infrequently accessed |
| **API Name** | standard |
| **Volume Size** | 1 GiB-1 TiB |
| **Max. IOPS/Volume** | 40–200 |
| **Max. Throughput/Volume** | 40–90 MiB/s |
| **Max. IOPS/Instance** | 80,000 |
| **Max. Throughput/Instance** | 1,750 MiB/s |
| **Dominant Performance Attribute** | IOPS |

**Note**

Linux AMIs require GPT partition tables and GRUB 2 for boot volumes 2 TiB (2048 GiB) or larger. Many Linux AMIs today use the MBR partitioning scheme, which only supports up to 2047 GiB boot volumes. If your instance does not boot with a boot volume that is 2 TiB or larger, the AMI you are using may be limited to a 2047 GiB boot volume size. Non-boot volumes do not have this limitation on Linux instances.

There are several factors that can affect the performance of EBS volumes, such as instance configuration, I/O characteristics, and workload demand. For more information about getting the most out of your EBS volumes, see [Amazon EBS Volume Performance on Linux Instances](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSPerformance.html).

For more information about pricing for these volume types, see [Amazon EBS Pricing](https://aws.amazon.com/ebs/pricing/).

## General Purpose SSD (gp2) Volumes

General Purpose SSD (gp2) volumes offer cost-effective storage that is ideal for a broad range of workloads. These volumes deliver single-digit millisecond latencies and the ability to burst to 3,000 IOPS for extended periods of time. Between a minimum of 100 IOPS (at 33.33 GiB and below) and a maximum of 10,000 IOPS (at 3,334 GiB and above), baseline performance scales linearly at 3 IOPS per GiB of volume size. AWS designs gp2 volumes to deliver the provisioned performance 99% of the time. A gp2 volume can range in size from 1 GiB to 16 TiB.

### I/O Credits and Burst Performance

The performance of gp2 volumes is tied to volume size, which determines the baseline performance level of the volume and how quickly it accumulates I/O credits; larger volumes have higher baseline performance levels and accumulate I/O credits faster. I/O credits represent the available bandwidth that your gp2 volume can use to burst large amounts of I/O when more than the baseline performance is needed. The more credits your volume has for I/O, the more time it can burst beyond its baseline performance level and the better it performs when more performance is needed. The following diagram shows the burst-bucket behavior for gp2.


              gp2 burst bucket
            

Each volume receives an initial I/O credit balance of 5.4 million I/O credits, which is enough to sustain the maximum burst performance of 3,000 IOPS for 30 minutes. This initial credit balance is designed to provide a fast initial boot cycle for boot volumes and to provide a good bootstrapping experience for other applications. Volumes earn I/O credits at the baseline performance rate of 3 IOPS per GiB of volume size. For example, a 100 GiB gp2 volume has a baseline performance of 300 IOPS.


              Comparing baseline performance and burst IOPS
            

When your volume requires more than the baseline performance I/O level, it draws on I/O credits in the credit balance to burst to the required performance level, up to a maximum of 3,000 IOPS. Volumes larger than 1,000 GiB have a baseline performance that is equal or greater than the maximum burst performance, and their I/O credit balance never depletes. When your volume uses fewer I/O credits than it earns in a second, unused I/O credits are added to the I/O credit balance. The maximum I/O credit balance for a volume is equal to the initial credit balance (5.4 million I/O credits).

The following table lists several volume sizes and the associated baseline performance of the volume (which is also the rate at which it accumulates I/O credits), the burst duration at the 3,000 IOPS maximum (when starting with a full credit balance), and the time in seconds that the volume would take to refill an empty credit balance.

|  |  |  |  |
| --- | --- | --- | --- |
| **Volume size (GiB)** | **Baseline performance (IOPS)** | **Maximum burst duration @ 3,000 IOPS (seconds)** | **Seconds to fill empty credit balance** |
| 1 | 100 | 1862 | 54,000 |
| 100 | 300 | 2,000 | 18,000 |
| 214 (Min. size for max. throughput) | 642 | 2,290 | 8,412 |
| 250 | 750 | 2,400 | 7,200 |
| 500 | 1,500 | 3,600 | 3,600 |
| 750 | 2,250 | 7,200 | 2,400 |
| 1,000 | 3,000 | N/A\* | N/A\* |
| 3,334 (Min. size for max. IOPS) | 10,000 | N/A\* | N/A\* |
| 16,384 (16 TiB, max. volume size) | 10,000 | N/A\* | N/A\* |

\* Bursting and I/O credits are only relevant to volumes under 1,000 GiB, where burst performance exceeds baseline performance.

The burst duration of a volume is dependent on the size of the volume, the burst IOPS required, and the credit balance when the burst begins. This is shown in the following equation:

(Credit balance)

Burst duration = ------------------------------------

(Burst IOPS) - 3(Volume size in GiB)

**What happens if I empty my I/O credit balance?**

If your gp2 volume uses all of its I/O credit balance, the maximum IOPS performance of the volume remains at the baseline IOPS performance level (the rate at which your volume earns credits) and the volume's maximum throughput is reduced to the baseline IOPS multiplied by the maximum I/O size. Throughput can never exceed 160 MiB/s. When I/O demand drops below the baseline level and unused credits are added to the I/O credit balance, the maximum IOPS performance of the volume again exceeds the baseline. For example, a 100 GiB gp2 volume with an empty credit balance has a baseline performance of 300 IOPS and a throughput limit of 75 MiB/s (300 I/O operations per second \* 256 KiB per I/O operation = 75 MiB/s). The larger a volume is, the greater the baseline performance is and the faster it replenishes the credit balance. For more information about how IOPS are measured, see [I/O Characteristics](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-io-characteristics.html).

If you notice that your volume performance is frequently limited to the baseline level (due to an empty I/O credit balance), you should consider using a larger gp2 volume (with a higher baseline performance level) or switching to an io1 volume for workloads that require sustained IOPS performance greater than 10,000 IOPS.

For information about using CloudWatch metrics and alarms to monitor your burst bucket balance, see [Monitoring the Burst Bucket Balance for gp2, st1, and sc1 Volumes](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSVolumeTypes.html#monitoring_burstbucket).

### Throughput Performance

Throughput for a gp2 volume can be calculated using the following formula, up to the throughput limit of 160 MiB/s:

(Volume size in GiB) × (IOPS per GiB) × (I/O size in KiB) = Throughput in MiB/s

Therefore the smallest volume size that achieves the maximum throughput is given by:

(160 MiB/s)

----------- (3 IOPS/GiB) = 214 GiB

(256 KiB)

## Provisioned IOPS SSD (io1) Volumes

Provisioned IOPS SSD (io1) volumes are designed to meet the needs of I/O-intensive workloads, particularly database workloads, that are sensitive to storage performance and consistency. Unlike gp2, which uses a bucket and credit model to calculate performance, an io1 volume allows you to specify a consistent IOPS rate when you create the volume, and Amazon EBS delivers within 10 percent of the provisioned IOPS performance 99.9 percent of the time over a given year.

An io1 volume can range in size from 4 GiB to 16 TiB and you can provision 100 up to 32,000 IOPS per volume. The maximum ratio of provisioned IOPS to requested volume size (in GiB) is 50:1. For example, a 100 GiB volume can be provisioned with up to 5,000 IOPS. Any volume 640 GiB in size or greater allows provisioning up to the 32,000 IOPS maximum (50 × 640 GiB = 32,000).

The throughput limit of io1 volumes is 256 KiB/s for each IOPS provisioned, up to a maximum of 500 MiB/s (at 32,000 IOPS).


            Throughput limits for io1 volumes
          

Your per-I/O latency experience depends on the IOPS provisioned and your workload pattern. For the best per-I/O latency experience, we recommend that you provision an IOPS-to-GiB ratio greater than 2:1. For example, a 2,000 IOPS volume should be smaller than 1,000 GiB.

**Note**

Some AWS accounts created before 2012 might have access to Availability Zones in us-west-1 or ap-northeast-1 that do not support Provisioned IOPS SSD (io1) volumes. If you are unable to create an io1 volume (or launch an instance with an io1 volume in its block device mapping) in one of these regions, try a different Availability Zone in the region. You can verify that an Availability Zone supports io1 volumes by creating a 4 GiB io1 volume in that zone.

## Throughput Optimized HDD (st1) Volumes

Throughput Optimized HDD (st1) volumes provide low-cost magnetic storage that defines performance in terms of throughput rather than IOPS. This volume type is a good fit for large, sequential workloads such as Amazon EMR, ETL, data warehouses, and log processing. Bootable st1 volumes are not supported.

Throughput Optimized HDD (st1) volumes, though similar to Cold HDD (sc1) volumes, are designed to support frequently accessed data.

This volume type is optimized for workloads involving large, sequential I/O, and we recommend that customers with workloads performing small, random I/O use gp2. For more information, see [**Inefficiency of Small Read/Writes on HDD**](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSVolumeTypes.html#inefficiency).

### Throughput Credits and Burst Performance

Like gp2, st1 uses a burst-bucket model for performance. Volume size determines the baseline throughput of your volume, which is the rate at which the volume accumulates throughput credits. Volume size also determines the burst throughput of your volume, which is the rate at which you can spend credits when they are available. Larger volumes have higher baseline and burst throughput. The more credits your volume has, the longer it can drive I/O at the burst level.

The following diagram shows the burst-bucket behavior for st1.


              st1 burst bucket
            

Subject to throughput and throughput-credit caps, the available throughput of an st1 volume is expressed by the following formula:

(Volume size) x (Credit accumulation rate per TiB) = Throughput

For a 1-TiB st1 volume, burst throughput is limited to 250 MiB/s, the bucket fills with credits at 40 MiB/s, and it can hold up to 1 TiB-worth of credits.

Larger volumes scale these limits linearly, with throughput capped at a maximum of 500 MiB/s. After the bucket is depleted, throughput is limited to the baseline rate of 40 MiB/s per TiB.

On volume sizes ranging from 0.5 to 16 TiB, baseline throughput varies from 20 to a cap of 500 MiB/s, which is reached at 12.5 TiB as follows:

40 MiB/s

12.5 TiB x ---------- = 500 MiB/s

1 TiB

Burst throughput varies from 125 MiB/s to a cap of 500 MiB/s, which is reached at 2 TiB as follows:

250 MiB/s

2 TiB x ---------- = 500 MiB/s

1 TiB

The following table states the full range of base and burst throughput values for st1:

|  |  |  |
| --- | --- | --- |
| **Volume Size (TiB)** | **ST1 Base Throughput (MiB/s)** | **ST1 Burst Throughput (MiB/s)** |
| 0.5 | 20 | 125 |
| 1 | 40 | 250 |
| 2 | 80 | 500 |
| 3 | 120 | 500 |
| 4 | 160 | 500 |
| 5 | 200 | 500 |
| 6 | 240 | 500 |
| 7 | 280 | 500 |
| 8 | 320 | 500 |
| 9 | 360 | 500 |
| 10 | 400 | 500 |
| 11 | 440 | 500 |
| 12 | 480 | 500 |
| 12.5 | 500 | 500 |
| 13 | 500 | 500 |
| 14 | 500 | 500 |
| 15 | 500 | 500 |
| 16 | 500 | 500 |

The following diagram plots the table values:


              Comparing st1 base and burst throughput
            

**Note**

When you create a snapshot of a Throughput Optimized HDD (st1) volume, performance may drop as far as the volume's baseline value while the snapshot is in progress.

For information about using CloudWatch metrics and alarms to monitor your burst bucket balance, see [Monitoring the Burst Bucket Balance for gp2, st1, and sc1 Volumes](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSVolumeTypes.html#monitoring_burstbucket).

## Cold HDD (sc1) Volumes

Cold HDD (sc1) volumes provide low-cost magnetic storage that defines performance in terms of throughput rather than IOPS. With a lower throughput limit than st1, sc1 is a good fit ideal for large, sequential cold-data workloads. If you require infrequent access to your data and are looking to save costs, sc1 provides inexpensive block storage. Bootable sc1 volumes are not supported.

Cold HDD (sc1) volumes, though similar to Throughput Optimized HDD (st1) volumes, are designed to support infrequently accessed data.

**Note**

This volume type is optimized for workloads involving large, sequential I/O, and we recommend that customers with workloads performing small, random I/O use gp2. For more information, see [**Inefficiency of Small Read/Writes on HDD**](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSVolumeTypes.html#inefficiency).

### Throughput Credits and Burst Performance

Like gp2, sc1 uses a burst-bucket model for performance. Volume size determines the baseline throughput of your volume, which is the rate at which the volume accumulates throughput credits. Volume size also determines the burst throughput of your volume, which is the rate at which you can spend credits when they are available. Larger volumes have higher baseline and burst throughput. The more credits your volume has, the longer it can drive I/O at the burst level.


              sc1 burst bucket
            

Subject to throughput and throughput-credit caps, the available throughput of an sc1 volume is expressed by the following formula:

(Volume size) x (Credit accumulation rate per TiB) = Throughput

For a 1-TiB sc1 volume, burst throughput is limited to 80 MiB/s, the bucket fills with credits at 12 MiB/s, and it can hold up to 1 TiB-worth of credits.

Larger volumes scale these limits linearly, with throughput capped at a maximum of 250 MiB/s. After the bucket is depleted, throughput is limited to the baseline rate of 12 MiB/s per TiB.

On volume sizes ranging from 0.5 to 16 TiB, baseline throughput varies from 6 MiB/s to a maximum of 192 MiB/s, which is reached at 16 TiB as follows:

12 MiB/s

16 TiB x ---------- = 192 MiB/s

1 TiB

Burst throughput varies from 40 MiB/s to a cap of 250 MiB/s, which is reached at 3.125 TiB as follows:

80 MiB/s

3.125 TiB x ----------- = 250 MiB/s

1 TiB

The following table states the full range of base and burst throughput values for sc1:

|  |  |  |
| --- | --- | --- |
| **Volume Size (TiB)** | **SC1 Base Throughput (MiB/s)** | **SC1 Burst Throughput (MiB/s)** |
| 0.5 | 6 | 40 |
| 1 | 12 | 80 |
| 2 | 24 | 160 |
| 3 | 36 | 240 |
| 3.125 | 37.5 | 250 |
| 4 | 48 | 250 |
| 5 | 60 | 250 |
| 6 | 72 | 250 |
| 7 | 84 | 250 |
| 8 | 96 | 250 |
| 9 | 108 | 250 |
| 10 | 120 | 250 |
| 11 | 132 | 250 |
| 12 | 144 | 250 |
| 13 | 156 | 250 |
| 14 | 168 | 250 |
| 15 | 180 | 250 |
| 16 | 192 | 250 |

The following diagram plots the table values:


              Comparing sc1 base and burst throughput
            

**Note**

When you create a snapshot of a Cold HDD (sc1) volume, performance may drop as far as the volume's baseline value while the snapshot is in progress.

For information about using CloudWatch metrics and alarms to monitor your burst bucket balance, see [Monitoring the Burst Bucket Balance for gp2, st1, and sc1 Volumes](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSVolumeTypes.html#monitoring_burstbucket).

## Magnetic (standard)

Magnetic volumes are backed by magnetic drives and are suited for workloads where data is accessed infrequently, and scenarios where low-cost storage for small volume sizes is important. These volumes deliver approximately 100 IOPS on average, with burst capability of up to hundreds of IOPS, and they can range in size from 1 GiB to 1 TiB.

**Note**

Magnetic is a Previous Generation Volume. For new applications, we recommend using one of the newer volume types. For more information, see [Previous Generation Volumes](https://aws.amazon.com/ebs/previous-generation/).

For information about using CloudWatch metrics and alarms to monitor your burst bucket balance, see [Monitoring the Burst Bucket Balance for gp2, st1, and sc1 Volumes](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSVolumeTypes.html#monitoring_burstbucket).

## Performance Considerations When Using HDD Volumes

For optimal throughput results using HDD volumes, plan your workloads with the following considerations in mind.

### Throughput Optimized HDD vs. Cold HDD

The st1 and sc1 bucket sizes vary according to volume size, and a full bucket contains enough tokens for a full volume scan. However, larger st1 and sc1 volumes take longer for the volume scan to complete due to per-instance and per-volume throughput limits. Volumes attached to smaller instances are limited to the per-instance throughput rather than the st1 or sc1 throughput limits.

Both st1 and sc1 are designed for performance consistency of 90% of burst throughput 99% of the time. Non-compliant periods are approximately uniformly distributed, targeting 99% of expected total throughput each hour.

The following table shows ideal scan times for volumes of various size, assuming full buckets and sufficient instance throughput.

In general, scan times are expressed by this formula:

Volume size

------------- = Scan time

Throughput

For example, taking the performance consistency guarantees and other optimizations into account, an st1customer with a 5-TiB volume can expect to complete a full volume scan in 2.91 to 3.27 hours.

5 TiB 5 TiB

----------- = ------------------- = 10,486 s = 2.91 hours (optimal)

500 MiB/s 0.00047684 TiB/s

2.91 hours

2.91 hours + -------------- = 3.27 hours (minimum expected)

(0.90)(0.99) <-- From expected performance of 90% of burst 99% of the time

Similarly, an sc1 customer with a 5-TiB volume can expect to complete a full volume scan in 5.83 to 6.54 hours.

5 TiB

------------------- = 20972 s = 5.83 hours (optimal)

0.000238418 TiB/s

5.83 hours

-------------- = 6.54 hours (minimum expected)

(0.90)(0.99)

|  |  |  |
| --- | --- | --- |
| **Volume Size (TiB)** | **ST1 Scan Time with Burst (Hours)\*** | **SC1 Scan Time with Burst (Hours)\*** |
| 1 | 1.17 | 3.64 |
| 2 | 1.17 | 3.64 |
| 3 | 1.75 | 3.64 |
| 4 | 2.33 | 4.66 |
| 5 | 2.91 | 5.83 |
| 6 | 3.50 | 6.99 |
| 7 | 4.08 | 8.16 |
| 8 | 4.66 | 9.32 |
| 9 | 5.24 | 10.49 |
| 10 | 5.83 | 11.65 |
| 11 | 6.41 | 12.82 |
| 12 | 6.99 | 13.98 |
| 13 | 7.57 | 15.15 |
| 14 | 8.16 | 16.31 |
| 15 | 8.74 | 17.48 |
| 16 | 9.32 | 18.64 |

\* These scan times assume an average queue depth (rounded to the nearest whole number) of four or more when performing 1 MiB of sequential I/O.

Therefore if you have a throughput-oriented workload that needs to complete scans quickly (up to 500 MiB/s), or requires several full volume scans a day, use st1. If you are optimizing for cost, your data is relatively infrequently accessed, and you don’t need more than 250 MiB/s of scanning performance, then use sc1.

### Inefficiency of Small Read/Writes on HDD

The performance model for st1 and sc1 volumes is optimized for sequential I/Os, favoring high-throughput workloads, offering acceptable performance on workloads with mixed IOPS and throughput, and discouraging workloads with small, random I/O.

For example, an I/O request of 1 MiB or less counts as a 1 MiB I/O credit. However, if the I/Os are sequential, they are merged into 1 MiB I/O blocks and count only as a 1 MiB I/O credit.

### Limitations on per-Instance Throughput

Throughput for st1 and sc1 volumes is always determined by the smaller of the following:

* Throughput limits of the volume
* Throughput limits of the instance

As for all Amazon EBS volumes, we recommend that you select an appropriate EBS-optimized EC2 instance in order to avoid network bottlenecks. For more information, see [Amazon EBS-Optimized Instances](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSOptimized.html).

## Monitoring the Burst Bucket Balance for gp2, st1, and sc1 Volumes

You can monitor the burst-bucket level for gp2, st1, and sc1 volumes using the EBS BurstBalance metric available in Amazon CloudWatch. This metric shows the percentage of I/O credits (for gp2) or throughput credits (for st1 and sc1) remaining in the burst bucket. For more information about the BurstBalancemetric and other metrics related to I/O, see [I/O Characteristics and Monitoring](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-io-characteristics.html). CloudWatch also allows you to set an alarm that notifies you when the BurstBalance value falls to a certain level. For more information, see [Creating Amazon CloudWatch Alarms](http://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/AlarmThatSendsEmail.html).

# Initializing Amazon EBS Volumes

New EBS volumes receive their maximum performance the moment that they are available and do not require initialization (formerly known as pre-warming). However, storage blocks on volumes that were restored from snapshots must be initialized (pulled down from Amazon S3 and written to the volume) before you can access the block. This preliminary action takes time and can cause a significant increase in the latency of an I/O operation the first time each block is accessed. For most applications, amortizing this cost over the lifetime of the volume is acceptable. Performance is restored after the data is accessed once.

You can avoid this performance hit in a production environment by reading from all of the blocks on your volume before you use it; this process is called initialization. For a new volume created from a snapshot, you should read all the blocks that have data before using the volume.

**Important**

While initializing io1 volumes that were restored from snapshots, the performance of the volume may drop below 50 percent of its expected level, which causes the volume to display a warning state in the **I/O Performance** status check. This is expected, and you can ignore the warning state on io1 volumes while you are initializing them. For more information, see [Monitoring Volumes with Status Checks](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/monitoring-volume-status.html#monitoring-volume-checks).

## Initializing Amazon EBS Volumes on Linux

New EBS volumes receive their maximum performance the moment that they are available and do not require initialization (formerly known as pre-warming). For volumes that have been restored from snapshots, use the **dd** or **fio** utilities to read from all of the blocks on a volume. All existing data on the volume will be preserved.

**To initialize a volume restored from a snapshot on Linux**

1. Attach the newly-restored volume to your Linux instance.
2. Use the **lsblk** command to list the block devices on your instance.
3. [ec2-user ~]$ **lsblk**
4. NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
5. xvdf 202:80 0 30G 0 disk

xvda1 202:1 0 8G 0 disk /

Here you can see that the new volume, /dev/xvdf, is attached, but not mounted (because there is no path listed under the MOUNTPOINT column).

1. Use the **dd** or **fio** utilities to read all of the blocks on the device. The **dd** command is installed by default on Linux systems, but **fio** is considerably faster because it allows multi-threaded reads.

**Note**

This step may take several minutes up to several hours, depending on your EC2 instance bandwidth, the IOPS provisioned for the volume, and the size of the volume.

[**dd**] The if (input file) parameter should be set to the drive you wish to initialize. The of (output file) parameter should be set to the Linux null virtual device, /dev/null. The bs parameter sets the block size of the read operation; for optimal performance, this should be set to 1 MB.

**Important**

Incorrect use of **dd** can easily destroy a volume's data. Be sure to follow precisely the example command below. Only the if=/dev/*xvdf* parameter will vary depending on the name of the device you are reading.

[ec2-user ~]$ **sudo dd if=/dev/*xvdf* of=/dev/null bs=1M**

[**fio**] If you have **fio** installed on your system, use the following command initialize your volume. The --filename (input file) parameter should be set to the drive you wish to initialize.

[ec2-user ~]$ **sudo fio --filename=/dev/*xvdf* --rw=read --bs=128k --iodepth=32 --ioengine=libaio --direct=1 --name=volume-initialize**

To install **fio** on Amazon Linux, use the following command:

sudo yum install -y fio

To install **fio** on Ubuntu, use the following command:

sudo apt-get install -y fio

When the operation is finished, you will see a report of the read operation. Your volume is now ready for use. For more information, see [Making an Amazon EBS Volume Available for Use on Linux](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-using-volumes.html).

# Monitoring the Status of Your Volumes

Amazon Web Services (AWS) automatically provides data, such as Amazon CloudWatch metrics and volume status checks, that you can use to monitor your Amazon Elastic Block Store (Amazon EBS) volumes.

## Monitoring Volumes with CloudWatch

CloudWatch metrics are statistical data that you can use to view, analyze, and set alarms on the operational behavior of your volumes.

The following table describes the types of monitoring data available for your Amazon EBS volumes.

|  |  |
| --- | --- |
| **Type** | **Description** |
| Basic | Data is available automatically in 5-minute periods at no charge. This includes data for the root device volumes for EBS-backed instances. |
| Detailed | Provisioned IOPS SSD (io1) volumes automatically send one-minute metrics to CloudWatch. |

When you get data from CloudWatch, you can include a Period request parameter to specify the granularity of the returned data. This is different than the period that we use when we collect the data (5-minute periods). We recommend that you specify a period in your request that is equal to or larger than the collection period to ensure that the returned data is valid.

You can get the data using either the CloudWatch API or the Amazon EC2 console. The console takes the raw data from the CloudWatch API and displays a series of graphs based on the data. Depending on your needs, you might prefer to use either the data from the API or the graphs in the console.

### Amazon EBS Metrics

Amazon Elastic Block Store (Amazon EBS) sends data points to CloudWatch for several metrics. Amazon EBS General Purpose SSD (gp2), Throughput Optimized HDD (st1) , Cold HDD (sc1), and Magnetic (standard) volumes automatically send five-minute metrics to CloudWatch. Provisioned IOPS SSD (io1) volumes automatically send one-minute metrics to CloudWatch. Data is only reported to CloudWatch when the volume is attached to an instance. For more information about how to monitor Amazon EBS, see [Monitoring the Status of Your Volumes](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/monitoring-volume-status.html) in the Amazon EC2 User Guide for Linux Instances.

The AWS/EBS namespace includes the following metrics.

|  |  |
| --- | --- |
| **Metric** | **Description** |
| VolumeReadBytes  VolumeWriteBytes | Provides information on the I/O operations in a specified period of time. The Sum statistic reports the total number of bytes transferred during the period. The Average statistic reports the average size of each I/O operation during the period, except on volumes attached to a C5, C5d, i3.metal, M5, or M5d instance, where the average represents the average over the specified period. The SampleCountstatistic reports the total number of I/O operations during the period, except on volumes attached to a C5, C5d, M5, or M5d instance, where the sample count represents the number of data points used in the statistical calculation. Data is reported to CloudWatch only when the volume is active.  The Minimum and Maximum statistics on this metric are supported only by volumes attached to a C5, C5d, i3.metal, M5, or M5d instance.  Units: Bytes |
| VolumeReadOps  VolumeWriteOps | The total number of I/O operations in a specified period of time.  To calculate the average I/O operations per second (IOPS) for the period, divide the total operations in the period by the number of seconds in that period.  The Minimum and Maximum statistics on this metric are supported only by volumes attached to a C5, C5d, i3.metal, M5, or M5d instance.  Units: Count |
| VolumeTotalReadTime  VolumeTotalWriteTime | The total number of seconds spent by all operations that completed in a specified period of time. If multiple requests are submitted at the same time, this total could be greater than the length of the period. For example, for a period of 5 minutes (300 seconds): if 700 operations completed during that period, and each operation took 1 second, the value would be 700 seconds.  The Average statistic on this metric is not relevant for volumes attached to a C5, C5d, i3.metal, M5, or M5d instance.  The Minimum and Maximum statistics on this metric are supported only by volumes attached to a C5, C5d, i3.metal, M5, or M5d instance.  Units: Seconds |
| VolumeIdleTime | The total number of seconds in a specified period of time when no read or write operations were submitted.  The Average statistic on this metric is not relevant for volumes attached to a C5, C5d, i3.metal, M5, or M5d instance.  The Minimum and Maximum statistics on this metric are supported only by volumes attached to a C5, C5d, i3.metal, M5, or M5d instance.  Units: Seconds |
| VolumeQueueLength | The number of read and write operation requests waiting to be completed in a specified period of time.  The Sum statistic on this metric is not relevant for volumes attached to a C5, C5d, i3.metal, M5, or M5d instance.  The Minimum and Maximum statistics on this metric are supported only by volumes attached to a C5, C5d, i3.metal, M5, or M5d instance.  Units: Count |
| VolumeThroughputPercentage | Used with Provisioned IOPS SSD volumes only. The percentage of I/O operations per second (IOPS) delivered of the total IOPS provisioned for an Amazon EBS volume. Provisioned IOPS SSD volumes deliver within 10 percent of the provisioned IOPS performance 99.9 percent of the time over a given year.  During a write, if there are no other pending I/O requests in a minute, the metric value will be 100 percent. Also, a volume's I/O performance may become degraded temporarily due to an action you have taken (for example, creating a snapshot of a volume during peak usage, running the volume on a non-EBS-optimized instance, or accessing data on the volume for the first time).  Units: Percent |
| VolumeConsumedReadWriteOps | Used with Provisioned IOPS SSD volumes only. The total amount of read and write operations (normalized to 256K capacity units) consumed in a specified period of time.  I/O operations that are smaller than 256K each count as 1 consumed IOPS. I/O operations that are larger than 256K are counted in 256K capacity units. For example, a 1024K I/O would count as 4 consumed IOPS.  Units: Count |
| BurstBalance | Used with General Purpose SSD (gp2), Throughput Optimized HDD (st1), and Cold HDD (sc1) volumes only. Provides information about the percentage of I/O credits (for gp2) or throughput credits (for st1and sc1) remaining in the burst bucket. Data is reported to CloudWatch only when the volume is active. If the volume is not attached, no data is reported.  The Sum statistic on this metric is not relevant for volumes attached to a C5, C5d, i3.metal, M5, or M5d instance.  Units: Percent |

### Dimensions for Amazon EBS Metrics

The only dimension that Amazon EBS sends to CloudWatch is the volume ID. This means that all available statistics are filtered by volume ID.

### Graphs in the Amazon EC2 Console

After you create a volume, you can view the volume's monitoring graphs in the Amazon EC2 console. Select a volume on the **Volumes** page in the console and choose **Monitoring**. The following table lists the graphs that are displayed. The column on the right describes how the raw data metrics from the CloudWatch API are used to produce each graph. The period for all the graphs is 5 minutes.

|  |  |
| --- | --- |
| **Graph** | **Description using raw metrics** |
| Read Bandwidth (KiB/s) | Sum(VolumeReadBytes) / Period / 1024 |
| Write Bandwidth (KiB/s) | Sum(VolumeWriteBytes) / Period / 1024 |
| Read Throughput (IOPS) | Sum(VolumeReadOps) / Period |
| Write Throughput (IOPS) | Sum(VolumeWriteOps) / Period |
| Avg Queue Length (Operations) | Avg(VolumeQueueLength) |
| % Time Spent Idle | Sum(VolumeIdleTime) / Period × 100 |
| Avg Read Size (KiB/Operation) | Avg(VolumeReadBytes) / 1024  **Note**  For C5, C5d, i3.metal, M5, and M5d instances, the following formula derives Average Read Size using [CloudWatch Metric Math](http://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/using-metric-math.html):  (Sum(VolumeReadBytes) / Sum(VolumeReadOps)) / 1024  The VolumeReadBytes and VolumeReadOps metrics are available in the EBS CloudWatch console. |
| Avg Write Size (KiB/Operation) | Avg(VolumeWriteBytes) / 1024  **Note**  For C5, C5d, i3.metal, M5, and M5d instances, the following formula derives Average Write Size using [CloudWatch Metric Math](http://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/using-metric-math.html):  (Sum(VolumeWriteBytes) / Sum(VolumeWriteOps)) / 1024  The VolumeWriteBytes and VolumeWriteOps metrics are available in the EBS CloudWatch console. |
| Avg Read Latency (ms/Operation) | Avg(VolumeTotalReadTime) × 1000  **Note**  For C5, C5d, i3.metal, M5, and M5d instances, the following formula derives Average Read Latency using [CloudWatch Metric Math](http://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/using-metric-math.html):  (Sum(VolumeTotalReadTime) / Sum(VolumeReadOps)) × 1000  The VolumeTotalReadTime and VolumeReadOps metrics are available in the EBS CloudWatch console. |
| Avg Write Latency (ms/Operation) | Avg(VolumeTotalWriteTime) × 1000  **Note**  For C5, C5d, i3.metal, M5, and M5d instances, the following formula derives Average Write Latency using [CloudWatch Metric Math](http://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/using-metric-math.html):  (Sum(VolumeTotalWriteTime) / Sum(VolumeWriteOps)) \* 1000  The VolumeTotalWriteTime and VolumeWriteOps metrics are available in the EBS CloudWatch console. |

For the average latency graphs and average size graphs, the average is calculated over the total number of operations (read or write, whichever is applicable to the graph) that completed during the period.

## Monitoring Volumes with Status Checks

Volume status checks enable you to better understand, track, and manage potential inconsistencies in the data on an Amazon EBS volume. They are designed to provide you with the information that you need to determine whether your Amazon EBS volumes are impaired, and to help you control how a potentially inconsistent volume is handled.

Volume status checks are automated tests that run every 5 minutes and return a pass or fail status. If all checks pass, the status of the volume is ok. If a check fails, the status of the volume is impaired. If the status is insufficient-data, the checks may still be in progress on the volume. You can view the results of volume status checks to identify any impaired volumes and take any necessary actions.

When Amazon EBS determines that a volume's data is potentially inconsistent, the default is that it disables I/O to the volume from any attached EC2 instances, which helps to prevent data corruption. After I/O is disabled, the next volume status check fails, and the volume status is impaired. In addition, you'll see an event that lets you know that I/O is disabled, and that you can resolve the impaired status of the volume by enabling I/O to the volume. We wait until you enable I/O to give you the opportunity to decide whether to continue to let your instances use the volume, or to run a consistency check using a command, such as **fsck**, before doing so.

**Note**

Volume status is based on the volume status checks, and does not reflect the volume state. Therefore, volume status does not indicate volumes in the error state (for example, when a volume is incapable of accepting I/O.)

If the consistency of a particular volume is not a concern for you, and you'd prefer that the volume be made available immediately if it's impaired, you can override the default behavior by configuring the volume to automatically enable I/O. If you enable the AutoEnableIO volume attribute, the volume status check continues to pass. In addition, you'll see an event that lets you know that the volume was determined to be potentially inconsistent, but that its I/O was automatically enabled. This enables you to check the volume's consistency or replace it at a later time.

The I/O performance status check compares actual volume performance to the expected performance of a volume and alerts you if the volume is performing below expectations. This status check is only available for io1 volumes that are attached to an instance and is not valid for General Purpose SSD (gp2), Throughput Optimized HDD (st1), Cold HDD (sc1), or Magnetic (standard) volumes. The I/O performance status check is performed once every minute and CloudWatch collects this data every 5 minutes, so it may take up to 5 minutes from the moment you attach a io1 volume to an instance for this check to report the I/O performance status.

**Important**

While initializing io1 volumes that were restored from snapshots, the performance of the volume may drop below 50 percent of its expected level, which causes the volume to display a warning state in the **I/O Performance** status check. This is expected, and you can ignore thewarning state on io1 volumes while you are initializing them. For more information, see [Initializing Amazon EBS Volumes](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-initialize.html).

The following table lists statuses for Amazon EBS volumes.

|  |  |  |
| --- | --- | --- |
| **Volume status** | **I/O enabled status** | **I/O performance status (only available for Provisioned IOPS volumes)** |
| ok | Enabled (I/O Enabled or I/O Auto-Enabled) | Normal (Volume performance is as expected) |
| warning | Enabled (I/O Enabled or I/O Auto-Enabled) | Degraded (Volume performance is below expectations)  Severely Degraded (Volume performance is well below expectations) |
| impaired | Enabled (I/O Enabled or I/O Auto-Enabled)  Disabled (Volume is offline and pending recovery, or is waiting for the user to enable I/O) | Stalled (Volume performance is severely impacted)  Not Available (Unable to determine I/O performance because I/O is disabled) |
| insufficient-data | Enabled (I/O Enabled or I/O Auto-Enabled)  Insufficient Data | Insufficient Data |

To view and work with status checks, you can use the Amazon EC2 console, the API, or the command line interface.

**To view status checks in the console**

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Volumes**.
3. On the **EBS Volumes** page, use the **Volume Status** column lists the operational status of each volume.
4. To view an individual volume's status, select the volume, and choose **Status Checks**.


                        Viewing EBS volume status
                    

1. If you have a volume with a failed status check (status is impaired), see [Working with an Impaired Volume](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/monitoring-volume-status.html#work_volumes_impaired).

Alternatively, you can use the **Events** pane to view all events for your instances and volumes in a single pane. For more information, see [Monitoring Volume Events](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/monitoring-volume-status.html#monitoring-vol-events).

**To view volume status information with the command line**

You can use one of the following commands to view the status of your Amazon EBS volumes. For more information about these command line interfaces, see [Accessing Amazon EC2](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html#access-ec2).

* [describe-volume-status](http://docs.aws.amazon.com/cli/latest/reference/ec2/describe-volume-status.html) (AWS CLI)
* [Get-EC2VolumeStatus](http://docs.aws.amazon.com/powershell/latest/reference/items/Get-EC2VolumeStatus.html) (AWS Tools for Windows PowerShell)

## Monitoring Volume Events

When Amazon EBS determines that a volume's data is potentially inconsistent, it disables I/O to the volume from any attached EC2 instances by default. This causes the volume status check to fail, and creates a volume status event that indicates the cause of the failure.

To automatically enable I/O on a volume with potential data inconsistencies, change the setting of the AutoEnableIO volume attribute. For more information about changing this attribute, see [Working with an Impaired Volume](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/monitoring-volume-status.html#work_volumes_impaired).

Each event includes a start time that indicates the time at which the event occurred, and a duration that indicates how long I/O for the volume was disabled. The end time is added to the event when I/O for the volume is enabled.

Volume status events include one of the following descriptions:

Awaiting Action: Enable IO

Volume data is potentially inconsistent. I/O is disabled for the volume until you explicitly enable it. The event description changes to **IO Enabled** after you explicitly enable I/O.

IO Enabled

I/O operations were explicitly enabled for this volume.

IO Auto-Enabled

I/O operations were automatically enabled on this volume after an event occurred. We recommend that you check for data inconsistencies before continuing to use the data.

Normal

For io1 volumes only. Volume performance is as expected.

Degraded

For io1 volumes only. Volume performance is below expectations.

Severely Degraded

For io1 volumes only. Volume performance is well below expectations.

Stalled

For io1 volumes only. Volume performance is severely impacted.

You can view events for your volumes using the Amazon EC2 console, the API, or the command line interface.

**To view events for your volumes in the console**

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Events**.
3. All instances and volumes that have events are listed. You can filter by volume to view only volume status. You can also filter on specific status types.
4. Select a volume to view its specific event.


                        Viewing volume events
                    

If you have a volume where I/O is disabled, see [Working with an Impaired Volume](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/monitoring-volume-status.html#work_volumes_impaired). If you have a volume where I/O performance is below normal, this might be a temporary condition due to an action you have taken (e.g., creating a snapshot of a volume during peak usage, running the volume on an instance that cannot support the I/O bandwidth required, accessing data on the volume for the first time, etc.).

**To view events for your volumes with the command line**

You can use one of the following commands to view event information for your Amazon EBS volumes. For more information about these command line interfaces, see [Accessing Amazon EC2](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html#access-ec2).

* [describe-volume-status](http://docs.aws.amazon.com/cli/latest/reference/ec2/describe-volume-status.html) (AWS CLI)
* [Get-EC2VolumeStatus](http://docs.aws.amazon.com/powershell/latest/reference/items/Get-EC2VolumeStatus.html) (AWS Tools for Windows PowerShell)

## Working with an Impaired Volume

This section discusses your options if a volume is impaired because the volume's data is potentially inconsistent.

**Options**

* [Option 1: Perform a Consistency Check on the Volume Attached to its Instance](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/monitoring-volume-status.html#work_volumes_impaired_option1)
* [Option 2: Perform a Consistency Check on the Volume Using Another Instance](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/monitoring-volume-status.html#work_volumes_impaired_option2)
* [Option 3: Delete the Volume If You No Longer Need It](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/monitoring-volume-status.html#work_volumes_impaired_option3)

### Option 1: Perform a Consistency Check on the Volume Attached to its Instance

The simplest option is to enable I/O and then perform a data consistency check on the volume while the volume is still attached to its Amazon EC2 instance.

**To perform a consistency check on an attached volume**

1. Stop any applications from using the volume.
2. Enable I/O on the volume.
   1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
   2. In the navigation pane, choose **Volumes**.
   3. Select the volume on which to enable I/O operations.
   4. In the details pane, choose **Enable Volume IO**.


                                    Enable IO
                                

* 1. In **Enable Volume IO**, choose **Yes, Enable**.

1. Check the data on the volume.
   1. Run the **fsck** command.
   2. (Optional) Review any available application or system logs for relevant error messages.
   3. If the volume has been impaired for more than 20 minutes you can contact support. Choose **Troubleshoot**, and then on the **Troubleshoot Status Checks** dialog box, choose **Contact Support** to submit a support case.

**To enable I/O for a volume with the command line**

You can use one of the following commands to view event information for your Amazon EBS volumes. For more information about these command line interfaces, see [Accessing Amazon EC2](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html#access-ec2).

* [enable-volume-io](http://docs.aws.amazon.com/cli/latest/reference/ec2/enable-volume-io.html) (AWS CLI)
* [Enable-EC2VolumeIO](http://docs.aws.amazon.com/powershell/latest/reference/items/Enable-EC2VolumeIO.html) (AWS Tools for Windows PowerShell)

### Option 2: Perform a Consistency Check on the Volume Using Another Instance

Use the following procedure to check the volume outside your production environment.

**Important**

This procedure may cause the loss of write I/Os that were suspended when volume I/O was disabled.

**To perform a consistency check on a volume in isolation**

1. Stop any applications from using the volume.
2. Detach the volume from the instance.
   1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
   2. In the navigation pane, choose **Volumes**.
   3. Select the volume to detach.
   4. Choose **Actions**, **Force Detach Volume**. You'll be prompted for confirmation.
3. Enable I/O on the volume.
   1. In the navigation pane, choose **Volumes**.
   2. Select the volume that you detached in the previous step.
   3. In the details pane, choose **Enable Volume IO**.


                                    Enable IO
                                

* 1. In the **Enable Volume IO** dialog box, choose **Yes, Enable**.

1. Attach the volume to another instance. For information, see [Launch Your Instance](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/LaunchingAndUsingInstances.html) and [Attaching an Amazon EBS Volume to an Instance](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-attaching-volume.html).
2. Check the data on the volume.
   1. Run the **fsck** command.
   2. (Optional) Review any available application or system logs for relevant error messages.
   3. If the volume has been impaired for more than 20 minutes, you can contact support. Choose **Troubleshoot**, and then in the troubleshooting dialog box, choose **Contact Support** to submit a support case.

**To enable I/O for a volume with the command line**

You can use one of the following commands to view event information for your Amazon EBS volumes. For more information about these command line interfaces, see [Accessing Amazon EC2](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html#access-ec2).

* [enable-volume-io](http://docs.aws.amazon.com/cli/latest/reference/ec2/enable-volume-io.html) (AWS CLI)
* [Enable-EC2VolumeIO](http://docs.aws.amazon.com/powershell/latest/reference/items/Enable-EC2VolumeIO.html) (AWS Tools for Windows PowerShell)

### Option 3: Delete the Volume If You No Longer Need It

If you want to remove the volume from your environment, simply delete it. For information about deleting a volume, see [Deleting an Amazon EBS Volume](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-deleting-volume.html).

If you have a recent snapshot that backs up the data on the volume, you can create a new volume from the snapshot. For information about creating a volume from a snapshot, see [Restoring an Amazon EBS Volume from a Snapshot](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-restoring-volume.html).

## Working with the AutoEnableIO Volume Attribute

When Amazon EBS determines that a volume's data is potentially inconsistent, it disables I/O to the volume from any attached EC2 instances by default. This causes the volume status check to fail, and creates a volume status event that indicates the cause of the failure. If the consistency of a particular volume is not a concern, and you prefer that the volume be made available immediately if it's impaired, you can override the default behavior by configuring the volume to automatically enable I/O. If you enable the AutoEnableIO volume attribute, I/O between the volume and the instance is automatically re-enabled and the volume's status check will pass. In addition, you'll see an event that lets you know that the volume was in a potentially inconsistent state, but that its I/O was automatically enabled. When this event occurs, you should check the volume's consistency and replace it if necessary. For more information, see [Monitoring Volume Events](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/monitoring-volume-status.html#monitoring-vol-events).

This section explains how to view and modify the AutoEnableIO attribute of a volume using the Amazon EC2 console, the command line interface, or the API.

**To view the AutoEnableIO attribute of a volume in the console**

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Volumes**.
3. Select the volume.
4. In the lower pane, choose **Status Checks**.
5. In the **Status Checks** tab, **Auto-Enable IO** displays the current setting for your volume, eitherEnabled or Disabled.


                        View Auto-Enable IO
                    

**To modify the AutoEnableIO attribute of a volume in the console**

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Volumes**.
3. Select the volume.
4. At the top of the **Volumes** page, choose **Actions**.
5. Choose **Change Auto-Enable IO Setting**.


                        Change Auto-Enable IO setting
                    

1. In the **Change Auto-Enable IO Setting** dialog box, select the **Auto-Enable Volume IO** option to automatically enable I/O for an impaired volume. To disable the feature, clear the option.


                        Modify Auto-Enable IO setting
                    

1. Choose **Save**.

Alternatively, instead of completing steps 4-6 in the previous procedure, choose **Status Checks**, **Edit**.

**To view or modify the AutoEnableIO attribute of a volume with the command line**

You can use one of the following commands to view the AutoEnableIO attribute of your Amazon EBS volumes. For more information about these command line interfaces, see [Accessing Amazon EC2](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html#access-ec2).

* [describe-volume-attribute](http://docs.aws.amazon.com/cli/latest/reference/ec2/describe-volume-attribute.html) (AWS CLI)
* [Get-EC2VolumeAttribute](http://docs.aws.amazon.com/powershell/latest/reference/items/Get-EC2VolumeAttribute.html) (AWS Tools for Windows PowerShell)

To modify the AutoEnableIO attribute of a volume, you can use one of the commands below.

* [modify-volume-attribute](http://docs.aws.amazon.com/cli/latest/reference/ec2/modify-volume-attribute.html) (AWS CLI)
* [Edit-EC2VolumeAttribute](http://docs.aws.amazon.com/powershell/latest/reference/items/Edit-EC2VolumeAttribute.html) (AWS Tools for Windows PowerShell)