

AWS Storage Services Overview

A Look at Storage Services Offered by AWS

November 2015



© 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Notices

This document is provided for informational purposes only. It represents AWS's current product offerings and practices as of the date of issue of this document, which are subject to change without notice. Customers are responsible for making their own independent assessment of the information in this document and any use of AWS's products or services, each of which is provided "as is" without warranty of any kind, whether express or implied. This document does not create any warranties, representations, contractual commitments, conditions or assurances from AWS, its affiliates, suppliers or licensors. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

Contents








Abstract	4
Introduction	4
Amazon S3	5
Amazon Glacier	11
Amazon EBS	15
Amazon EC2 Instance Storage	21
AWS Storage Gateway	26
AWS Import/Export Snowball	30
Amazon CloudFront	34
References and Further Reading	38

Abstract

Amazon Web Services (AWS) is a flexible, cost-effective, easy-to-use cloud computing platform. This whitepaper is designed to help architects and developers understand the different storage services and features available in the AWS cloud. We provide an overview of each storage service or feature and describe usage patterns, performance, durability and availability, scalability and elasticity, security, interfaces, and the cost model.

Introduction

Amazon Web Services (AWS) provides low-cost data storage with high durability and availability. AWS offers storage choices for backup, archiving, and disaster recovery, and also block, file, and object storage. In this whitepaper, we examine the following AWS cloud storage services and features.

	Amazon Simple Storage Service (Amazon S3)	A service that provides scalable storage in the cloud
	Amazon Glacier	A service that provides low-cost archive storage in the cloud
	Amazon Elastic Block Store (Amazon EBS)	A service that provides persistent block storage volumes for Amazon Elastic Compute Cloud (Amazon EC2) instances
	Amazon EC2 Instance Storage	Temporary block storage volumes for Amazon EC2 instances
	AWS Storage Gateway	A service that integrates on-premises IT environments with cloud storage
	AWS Import/Export Snowball	A feature of the AWS Import/Export service that enables large-volume data transfer
	Amazon CloudFront	A service that provides a global content delivery network (CDN)

Amazon S3

[Amazon Simple Storage Service \(Amazon S3\)](#) provides developers and IT teams secure, durable, highly scalable, object storage at a very low cost. Amazon S3 provides a simple web services interface that can be used to easily store and retrieve any amount of data, at any time, from anywhere on the web. You can write, read, and delete objects containing from 1 byte to 5 TB of data. Amazon S3 is highly scalable, allowing concurrent read or write access to data by many

separate clients or application threads. Amazon S3 offers a range of storage classes designed for different use cases including the following:

- Amazon S3 Standard, for general-purpose storage of frequently accessed data
- Amazon S3 Standard-Infrequent Access (Standard-IA), for long-lived, but less frequently accessed data

Usage Patterns

One very common use for Amazon S3 is storage and distribution of static web content and media. This content can be delivered directly from Amazon S3, because each object in Amazon S3 has a unique HTTP URL. Alternatively, Amazon S3 can serve as an origin store for a CDN, such as Amazon CloudFront. Because of the elasticity of Amazon S3, it works particularly well for hosting web content that requires the bandwidth to address extreme demand spikes. Also, because no storage provisioning is required, Amazon S3 works well for fast-growing websites hosting data intensive, user-generated content, such as video and photo sharing sites.

Amazon S3 is also frequently used to host entire static websites. Amazon S3 provides a low-cost, highly available, and highly scalable solution for these, including storage for static HTML files, images, videos, and client-side scripts in such formats as JavaScript.

Amazon S3 is also commonly used as a data store for computation and large-scale analytics, such as financial transaction analysis, clickstream analytics, and media transcoding. Because of the horizontal scalability of Amazon S3, you can access your data from multiple computing nodes concurrently without being constrained by a single connection.

Finally, Amazon S3 is often used as a highly durable, scalable, and secure solution for backup and archiving of critical data. You can easily move cold data to Amazon Glacier using lifecycle management rules on data stored in Amazon S3. You can also use [Amazon S3 cross-region replication](#) to automatically copy objects across S3 buckets in different AWS regions asynchronously, providing disaster recovery solutions for business continuity.¹

Amazon S3 doesn't suit all storage situations. Listed following are some storage needs for which you should consider other AWS storage options instead of Amazon S3.

Storage Need	Solution	Alternative AWS Services
File system	Amazon S3 uses a flat namespace and isn't meant to serve as a standalone, POSIX-compliant file system. Instead, consider using Amazon Elastic File System (Amazon EFS) as a file system.	Amazon Elastic File System (EFS)
Structured data with query	Amazon S3 doesn't offer query capabilities to retrieve specific objects. When you use Amazon S3, we expect that you know the exact bucket name and key for the files you want to retrieve from the service. Thus, Amazon S3 cannot be used as a database or search engine by itself. Instead, you can pair Amazon S3 with Amazon DynamoDB, Amazon CloudSearch, or Amazon Relational Data Service (Amazon RDS) to index and query metadata about Amazon S3 buckets and objects.	Amazon DynamoDB Amazon RDS Amazon CloudSearch
Rapidly changing data	Data that must be updated very frequently might be better served by storage solutions that take into account read and write latencies, such as Amazon EBS volumes, Amazon RDS, Amazon DynamoDB, or relational databases running on Amazon EC2.	Amazon EBS Amazon DynamoDB Amazon RDS
Archival data	Data that requires encrypted archival storage with infrequent read access with a long recovery time objective (RTO) can be stored in Amazon Glacier more cost-effectively.	Amazon Glacier
Dynamic website hosting	Although Amazon S3 is ideal for static content websites, dynamic websites that depend on database interaction or use server-side scripting should be hosted on Amazon EC2.	Amazon EC2

Performance

In scenarios where you use Amazon S3 from within Amazon EC2 in the same region, access to Amazon S3 from Amazon EC2 is designed to be fast. Amazon S3 is also designed so that server-side latencies are insignificant relative to Internet latencies. In addition, Amazon S3 is built to scale storage, requests, and numbers of users to support an extremely large number of web-scale applications. If you access Amazon S3 using multiple threads, multiple applications, or multiple clients concurrently, total Amazon S3 aggregate throughput typically scales to rates that far exceed what any single server can generate or consume.

To speed up the upload of large objects, you can perform [multipart upload](#) to upload a single objects as set of parts.² After all parts of your object are uploaded, Amazon S3 assembles these parts and creates the object. Using multipart upload, you can get improved throughput and quick recovery from any network issues. You can also pause and resume object uploads and start uploading an object in Amazon S3 while it is still being created.

To speed up access to relevant data, many developers pair Amazon S3 with a search engine such as Amazon CloudSearch or a database such as Amazon DynamoDB or Amazon RDS. In these scenarios, Amazon S3 stores the actual information, and the search engine or database serves as the repository for associated metadata (for example, the object name, size, keywords, and so on). Metadata in the database can easily be indexed and queried, making it very efficient to locate an object's reference by using a search engine or a database query. This result can be used to pinpoint and retrieve the object itself from Amazon S3.

Durability and Availability

Amazon S3 Standard storage and Standard-IA storage provide the highest level of data durability and availability on the AWS platform by automatically and synchronously storing your data across both multiple devices and multiple facilities within your selected geographical region. Error correction is built-in, and there are no single points of failure. Amazon S3 is designed to sustain the concurrent loss of data in two facilities, making it very well-suited to serve as the primary data storage for mission-critical data. In fact, Amazon S3 is designed for 99.999999999 percent (11 nines) durability per object and 99.99 percent availability over a one-year period.

Additionally, you have a choice of enabling cross-region replication on each Amazon S3 bucket. Once enabled, cross-region replication automatically copies objects across buckets in different AWS regions asynchronously, providing 11 nines of durability and 99.99 percent availability on both the source and destination Amazon S3 objects.

Scalability and Elasticity

Amazon S3 has been designed to offer a very high level of scalability and elasticity automatically. Unlike a typical file system that encounters issues when storing

large number of files in a directory, Amazon S3 supports a virtually unlimited number of files in any bucket. Also, unlike a disk drive that has a limit on the total amount of data that can be stored before you must partition the data across drives and/or servers, an Amazon S3 bucket can store a virtually unlimited number of bytes. You can store any number of objects (files) in a single bucket, and Amazon S3 will automatically manage scaling and distributing redundant copies of your information to other servers in other locations in the same region, all using Amazon's high-performance infrastructure.

Security

Amazon S3 is highly secure; it provides multiple mechanisms for fine-grained control of access to Amazon S3 resources, and it supports encryption.

You can manage access to Amazon S3 by granting other AWS accounts and users permission to perform the resource operations by writing an [access policy](#).³

You can protect Amazon S3 data at rest by using [server-side encryption](#), in which you request Amazon S3 to encrypt your object before it's written to disks in data centers and decrypt it when you download the object or by using [client-side encryption](#), in which you encrypt your data on the client side and upload the encrypted data to Amazon S3.^{4 5} You can protect the data in transit by using Secure Sockets Layer (SSL) or client-side encryption.

You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. Additionally you can add an optional layer of security by enabling [Multi-Factor Authentication \(MFA\) Delete](#) for a bucket.⁶ With this option enabled for a bucket, two forms of authentication are required to change the versioning state of the bucket or to permanently delete an object version: valid AWS account credentials plus a six-digit code (a single-use, time-based password) from a physical or virtual token device.

To track requests for access to your bucket, you can enable [access logging](#).⁷ Each access log record provides details about a single access request, such as the requester, bucket name, request time, request action, response status, and error code, if any. Access log information can be useful in security and access audits. It

can also help you learn about your customer base and understand your Amazon S3 bill.

Interfaces

Amazon S3 provides standards-based REST and SOAP web service APIs for both management and data operations. These APIs allow Amazon S3 objects to be stored in uniquely named buckets (top-level folders). Each object must have a unique object key (file name) that serves as an identifier for the object within that bucket. Although Amazon S3 is a web-based object store rather than a traditional file system, you can easily emulate a file system hierarchy (folder1/folder2/file) in Amazon S3 by creating object key names that correspond to the full path name of each file.

Most developers building applications on Amazon S3 use a higher-level toolkit or software development kit (SDK) that wraps the underlying REST API. AWS SDKs are available for Android, Browser, iOS, Java, .NET, Node.js, PHP, Python, Ruby, and GO. The integrated AWS Command Line Interface (AWS CLI) also provides a set of high-level, Linux-like Amazon S3 file commands for common operations, such as **ls**, **cp**, **mv**, **sync**, and so on. Using the AWS CLI for Amazon S3, you can perform recursive uploads and downloads using a single folder-level Amazon S3 command, and also perform parallel transfers. You can also use the AWS CLI for command-line access to the low-level Amazon S3 API. Using the AWS Management Console, you can easily create and manage Amazon S3 buckets, upload and download objects, and browse the contents of your Amazon S3 buckets using a simple web-based user interface.

Additionally, you can use Amazon S3 notification feature to receive notifications when certain events happen in your bucket. Currently, Amazon S3 can publish events such as when a new object is created or deleted or a RRS object is lost. Notifications can be issued to [Amazon Simple Notification Service \(SNS\)](#) topics, [Amazon Simple Queue Service \(SQS\)](#) queues, and [AWS Lambda](#) functions.^{8 9 10}

Cost Model

With Amazon S3, you pay only for the storage you actually use. There is no minimum fee and no setup cost. Amazon S3 has three pricing components: storage (per GB per month), data transfer in or out (per GB per month), and requests (per n thousand requests per month). For new customers, AWS provides

[the AWS Free Tier](#), which includes up to 5 GB of Amazon S3 storage, 20,000 get requests, 2,000 put requests, and 15 GB of data transfer out each month for one year, for free.¹¹ You can find pricing information at [the Amazon S3 pricing page](#).¹²

Amazon Glacier

[Amazon Glacier](#) is an extremely low-cost storage service that provides highly secure, durable, and flexible storage for data archiving and online backup. With Amazon Glacier, customers can reliably store their data for as little as \$0.01 per gigabyte per month. Amazon Glacier enables customers to offload the administrative burdens of operating and scaling storage to AWS, so that they don't have to worry about capacity planning, hardware provisioning, data replication, hardware failure detection and repair, or time-consuming hardware migrations.

You store data in Amazon Glacier as archives. An archive can represent a single file, or you can combine several files to be uploaded as a single archive. Retrieving archives from Amazon Glacier requires the initiation of a job. You organize your archives in vaults.

Amazon Glacier is designed for use with other Amazon web services. You can seamlessly move data between Amazon Glacier and Amazon S3 using S3 data lifecycle policies. You can also use AWS Import/Export services to accelerate moving large amounts of data into Amazon Glacier using portable storage devices for transport.

Usage Patterns

Organizations are using Amazon Glacier to support a number of use cases. These use cases include archiving offsite enterprise information, media assets, and research and scientific data, and also performing digital preservation and magnetic tape replacement.

Amazon Glacier doesn't suit all storage situations. Listed following are a few storage needs for which you should consider other AWS storage options instead of Amazon Glacier.

Storage Need	Solution	AWS Services
Rapidly changing data	Data that must be updated very frequently might be better served by a storage solution with lower read/write latencies, such as Amazon EBS, Amazon RDS, Amazon DynamoDB, or relational databases running on EC2.	Amazon EBS Amazon RDS Amazon DynamoDB Amazon EC2
Immediate access	Data stored in Amazon Glacier is not available immediately. Retrieval jobs typically require 3–5 hours to complete, so if you need immediate access to your data, Amazon S3 is a better choice.	Amazon S3

Performance

Amazon Glacier is a low-cost storage service designed to store data that is infrequently accessed and long-lived. Amazon Glacier retrieval jobs typically complete in 3 to 5 hours.

You can improve the upload experience for larger archives by using [multipart upload](#) for archives up to about 40 TB (the single archive limit).¹³ You can upload separate parts of a large archive independently, in any order and in parallel, to improve the upload experience for larger archives. You can even perform [range retrievals](#) on archives stored in Amazon Glacier, by specifying a range or portion of the archive.¹⁴ Specifying a range of bytes for a retrieval can help control bandwidth costs, manage your data downloads, and retrieve a targeted part of a large archive.

Durability and Availability

Amazon Glacier is designed to provide average annual durability of 99.999999999 percent (11 nines) for an archive. The service redundantly stores data in multiple facilities and on multiple devices within each facility. To increase durability, Amazon Glacier synchronously stores your data across multiple facilities before returning SUCCESS on uploading an archive. Unlike traditional systems, which can require laborious data verification and manual repair, Amazon Glacier performs regular, systematic data integrity checks and is built to be automatically self-healing.

Scalability and Elasticity

Amazon Glacier scales to meet growing and often unpredictable storage requirements. A single archive is limited to 40 TB in size, but there is no limit to

the total amount of data you can store in the service. Whether you're storing petabytes or gigabytes, Amazon Glacier automatically scales your storage up or down as needed.

Security

By default, only you can access your Amazon Glacier data. If other people need to access your data, you can set up data access control in Amazon Glacier by using the [AWS Identity and Access Management \(IAM\) service](#).¹⁵ To do so, you simply create an IAM policy that specifies which users within an account have rights to operations on a given vault.

Amazon Glacier uses server-side encryption to encrypt all data at rest. Amazon Glacier handles key management and key protection for you by using one of the strongest block ciphers available, 256-bit Advanced Encryption Standard (AES-256). Customers who want to manage their own keys can encrypt data prior to uploading it.

Amazon Glacier allows you to lock vaults where long-term records retention is mandated by regulations or compliance rules. You can set compliance controls on individual Amazon Glacier vaults and enforce these by using lockable policies. For example, you might specify controls such as “undeletable records” or “time-based data retention” in a Vault Lock policy and then lock the policy from future edits. Once locked, the policy becomes immutable, and Amazon Glacier enforces the prescribed controls to help achieve your compliance objectives.

To help monitor data access, Amazon Glacier is integrated with [Amazon CloudTrail](#), allowing any API calls made to Amazon Glacier in your AWS account to be captured and stored in log files that are delivered to an Amazon S3 bucket that you specify.¹⁶

Interfaces

There are two ways to use Amazon Glacier, each with its own interfaces. The Amazon Glacier API provides both management and data operations.

First, Amazon Glacier provides a native, standards-based REST web services interface, and also Java and .NET SDKs. You can use the AWS Management Console or Amazon Glacier API actions to create vaults to organize the archives

in Amazon Glacier. You can then use the Amazon Glacier API actions to upload and retrieve archives, to monitor the status of your jobs, and also to configure your vault to send you a notification through [Amazon Simple Notification Service \(Amazon SNS\)](#) when a job is complete.¹⁷

Second, Amazon Glacier can be used as a storage class in Amazon S3 by using object lifecycle management that provides automatic, policy-driven archiving from Amazon S3 to Amazon Glacier. You simply set one or more lifecycle rules for an Amazon S3 bucket, defining what objects should be transitioned to Amazon Glacier and when. You can specify an absolute or relative time period (including 0 days) after which the specified Amazon S3 objects should be transitioned to Amazon Glacier. The Amazon S3 API now includes a new RESTORE operation, and the retrieval process from Amazon Glacier using RESTORE takes the same 3 to 5 hours as other Amazon Glacier retrievals. Retrieval puts a copy of the retrieved object in Amazon S3 RRS storage for a specified retention period; the original archived object remains stored in Amazon Glacier. For more information on how to use Amazon Glacier from Amazon S3, see the [Object Lifecycle Management](#) section of the *Amazon S3 Developer Guide*.¹⁸

Note that when using Amazon Glacier as a storage class in Amazon S3, you use the Amazon S3 API, and that when using “native” Amazon Glacier, you use the Amazon Glacier API. Objects archived to Amazon Glacier by using Amazon S3 can only be listed and retrieved by using the Amazon S3 API or the AWS Management Console—you cannot see them as archives in an Amazon Glacier vault.

Cost Model

With Amazon Glacier, you pay only for what you use and there is no minimum fee. In normal use, Amazon Glacier has three pricing components: storage (per GB per month), data transfer out (per GB per month), and requests (per thousand UPLOAD and RETRIEVAL requests per month).

Note that Amazon Glacier is designed with the expectation that retrievals are infrequent and unusual, and data will be stored for extended periods of time. You can retrieve up to 5 percent of your average monthly storage (prorated daily) for free each month. If you retrieve more than this amount of data in a month, you are charged an additional (per GB) retrieval fee. A prorated charge (per GB) also

applies for items deleted prior to 90 days' passage. You can find pricing information at [the Amazon Glacier pricing page](#).¹⁹

Amazon EBS

[Amazon Elastic Block Store \(Amazon EBS\)](#) volumes provide durable block-level storage for use with Amazon EC2 instances (virtual machines). Amazon EBS volumes are off-instance storage that persists independently from the running life of a single Amazon EC2 instance. After an Amazon EBS volume is attached to an EC2 instance, you can use the EBS volume like a physical hard drive, typically by formatting it with the file system of your choice and using the file I/O interface provided by the instance operating system. For Amazon Machine Images (AMIs) that are backed by Amazon EBS, you can use an Amazon EBS volume to boot an Amazon EC2 instance. You can also attach multiple Amazon EBS volumes to a single Amazon EC2 instance. Note, however, that any single Amazon EBS volume can be attached to only one Amazon EC2 instance at any point in time.

Amazon EBS also provides the ability to create point-in-time snapshots of volumes, which are persisted to Amazon S3. These snapshots can be used as the starting point for new Amazon EBS volumes, and to protect data for long-term durability. The same snapshot can be used to instantiate as many volumes as you want. These snapshots can be copied across AWS regions, making it easier to leverage multiple AWS regions for geographical expansion, data center migration, and disaster recovery. Sizes for Amazon EBS volumes range from 1 GB to 16 TB, and are allocated in 1 GB increments.

Usage Patterns

Amazon EBS is meant for data that changes relatively frequently and requires long-term persistence. Amazon EBS is well-suited for use as the primary storage for a database or file system, or for any applications that require access to raw block-level storage. Of the different Amazon EBS volume types, Amazon EBS Magnetic is well-suited for infrequently accessed data. Amazon EBS General Purpose, which is solid-state drive (SSD)-backed, is well-suited for boot volumes and small to medium databases. Amazon EBS Provisioned IOPS (also SSD-

backed) is well-suited for use with I/O-intensive workloads, relational databases, and NoSQL databases.

Amazon EBS doesn't suit all storage situations. Listed following are some storage needs for which you should consider other AWS storage options instead of Amazon EBS.

Storage Need	Solution	AWS Services
Temporary storage	Consider using local instance store volumes for needs such as scratch disks, buffers, queues, and caches.	Amazon Local Instance Store
Highly durable storage	If you need very highly durable storage, use Amazon S3. Amazon S3 standard storage is designed for 99.999999999 percent annual durability per object. You can even decide to snapshot the EBS volumes. Such a snapshot then gets saved in Amazon S3, thus providing you the durability of Amazon S3.	Amazon S3
Static data or web content	If your data doesn't change that often, Amazon S3 might represent a more cost-effective and scalable solution for storing this fixed information. Also, web content served out of Amazon EBS requires a web server running on Amazon EC2; in contrast, you can deliver web content directly out of Amazon S3.	Amazon S3

Performance

As described preceding, Amazon EBS provides three volume types: Magnetic, General Purposed (SSD-backed) and Provisioned IOPS (SSD-backed) volumes. They differ in performance characteristics and pricing model, allowing you to tailor your storage performance and cost to the needs of your applications. You can attach and stripe across multiple volumes of any type to increase the I/O performance available to your Amazon EC2 applications.

Magnetic volumes offer cost-effective storage for applications with moderate or bursty I/O requirements. Magnetic volumes are designed to deliver approximately 100 input/output operations per second (IOPS) on average, with a best effort ability to burst to hundreds of IOPS. Magnetic volumes are also well-suited for use as boot volumes, where the burst capability provides fast instance startup times.

General Purpose (SSD) volumes offer cost-effective storage that is ideal for a broad range of workloads. These volumes deliver single-digit millisecond latencies, the ability to burst to 3,000 IOPS for extended periods of time, and a baseline performance of 3 IOPS/GB up to a maximum of 10,000 IOPS (at 3,334 GB). General Purpose (SSD) volumes can range in size from 1 GB to 16 TB. General Purpose (SSD) volumes have a throughput limit range of 128 MB/second for volumes less than or equal to 170 GB; for volumes over 170 GB, this limit increases at the rate of 128 MB/second per TB to a maximum of 160 MB/second (at 214 GB and larger).

Provisioned IOPS (SSD) volumes are designed to deliver predictable high performance for I/O-intensive workloads, such as database workloads that are sensitive to storage performance and consistency in random access I/O throughput. You specify an IOPS rate when creating a volume, and then Amazon EBS delivers within 10 percent of the provisioned IOPS performance 99.9 percent of the time over a given year. A Provisioned IOPS (SSD) volume can range in size from 4 GB to 16 TB, and you can provision up to 20,000 IOPS per volume. The ratio of IOPS provisioned to the volume size requested can be a maximum of 30; for example, a volume with 3,000 IOPS must be at least 100 GB in size. Provisioned IOPS (SSD) volumes have a throughput limit range of 256 KB for each IOPS provisioned, up to a maximum of 320 MB/second (at 1,280 IOPS).

Because Amazon EBS volumes are network-attached devices, other network I/O performed by an Amazon EC2 instance, and also the total load on the shared network, can affect the performance of individual Amazon EBS volumes. To enable your Amazon EC2 instances to fully utilize the Provisioned IOPS on an Amazon EBS volume, you can launch selected Amazon EC2 instance types as Amazon EBS-optimized instances. Amazon EBS-optimized instances deliver dedicated throughput between Amazon EC2 and Amazon EBS, with speeds between 500 Mbps and 4,000 Mbps depending on the instance type used. When attached to Amazon EBS-optimized instances, Provisioned IOPS volumes are designed to deliver within 10 percent of the Provisioned IOPS performance 99.9 percent of the time. Newly created EBS volumes receive their maximum performance the moment that they are available and do not require initialization (formerly known as prewarming). However, you must initialize the storage blocks on volumes that were restored from snapshots before you can access the block.²⁰

Using Amazon EC2 with Amazon EBS, you can take advantage of many of the same disk performance optimization techniques that you do with on-premises servers and storage. For example, by attaching multiple Amazon EBS volumes to a single Amazon EC2 instance, you can partition the total application I/O load by allocating one volume for database log data, one or more volumes for database file storage, and other volumes for file system data. Each separate Amazon EBS volume can be configured as Amazon EBS General Purpose (SSD), Provisioned IOPS (SSD), or Magnetic as needed. Alternatively, you can stripe your data across multiple similarly provisioned Amazon EBS volumes using RAID 0 (disk striping) or logical volume manager software, thus aggregating available IOPS, total volume throughput, and total volume size.

Durability and Availability

Amazon EBS volumes are designed to be highly available and reliable. Amazon EBS volume data is replicated across multiple servers in a single Availability Zone to prevent the loss of data from the failure of any single component. The durability of your Amazon EBS volume depends on both the size of your volume and the amount of data that has changed since your last snapshot. Amazon EBS snapshots are incremental, point-in-time backups, containing only the data blocks changed since the last snapshot. Amazon EBS volumes are designed for an annual failure rate (AFR) of between 0.1 and 0.2 percent, where failure refers to a complete or partial loss of the volume, depending on the size and performance of the volume. This AFR makes EBS volumes 20 times more reliable than typical commodity disk drives, which fail with an AFR of around 4 percent. EBS also supports a snapshot feature, which is a good way to take point-in-time backups of your data.

To maximize both durability and availability of Amazon EBS data, you should create snapshots of your Amazon EBS volumes frequently. (For data consistency, we recommend briefly pausing any write operations to the volume, or unmounting the volume, while the snapshot command is issued. You can then safely continue to use the volume while the snapshot is pending completion.) If your Amazon EBS volume does fail, all snapshots of that volume remain intact, and you can recreate your volume from the last snapshot point. Because an Amazon EBS volume is created in a particular Availability Zone, the volume will be unavailable if the Availability Zone itself is unavailable. An Amazon EBS snapshot of a volume, however, is available across all the Availability Zones within a region, and you can use an Amazon EBS snapshot to create one or more

new Amazon EBS volumes in any Availability Zone in the region. Amazon EBS snapshots can also be copied from one region to another, and can easily be shared with other user accounts. Thus, Amazon EBS snapshots provides an easy-to-use disk clone or disk image mechanism for backup, sharing, and disaster recovery.

Scalability and Elasticity

Using the AWS Management Console or the Amazon EBS API, you can easily and rapidly provision and release Amazon EBS volumes to scale in and out with your total storage demands. Although individual Amazon EBS volumes cannot be resized, if you find that you need additional storage you have two ways to expand the amount of Amazon EBS space available for your Amazon EC2 instance.

The simplest approach is to create and attach a new Amazon EBS volume and begin using it together with your existing ones. However, if you need to expand the size of a single Amazon EBS volume, you can effectively resize a volume using a snapshot as described following:

1. Detach the original Amazon EBS volume.
2. Create a snapshot of the original Amazon EBS volume's data in Amazon S3.
3. Create a new Amazon EBS volume from the snapshot, but specify a larger size than the original volume.
4. Attach the new, larger volume to your Amazon EC2 instance in place of the original. (In many cases, an OS-level utility must also be used to expand the file system.)
5. Delete the original Amazon EBS volume.

Security

IAM enables access control for your Amazon EBS volumes, allowing you to specify who can access which EBS volumes.

Amazon EBS encryption offers seamless encryption of EBS data volumes and snapshots, eliminating the need to build and manage a secure key management

infrastructure. EBS encryption enables data-at-rest security by encrypting your data volumes and snapshots using Amazon-managed keys or keys that you create and manage using the AWS Key Management Service (AWS KMS). In addition, the encryption occurs on the servers that host Amazon EC2 instances, providing encryption of data as it moves between EC2 instances and EBS data volumes.

Access control plus encryption offers a strong defense-in-depth security strategy for your data. For more information, see [Amazon EBS Encryption](#) in the *Amazon EBS User Guide*.²¹

Interfaces

Amazon offers a REST management API for Amazon EBS, as well as support for EBS operations within both the AWS SDKs and the AWS Command Line Interface (CLI). The API actions and EBS operations are used to create, delete, describe, attach, and detach Amazon EBS volumes for your Amazon EC2 instances; to create, delete, and describe snapshots from Amazon EBS to Amazon S3; and to copy snapshots from one region to another. If you prefer to work with a graphical user interface, the AWS Management Console gives you all the capabilities of the API in an easy-to-use browser interface. Regardless of how you create your Amazon EBS volume, note that all storage is allocated at the time of volume creation, and that you are charged for this allocated storage even if you don't write data to it.

Amazon EBS doesn't provide a data API. Instead, Amazon EBS presents a block-device interface to the Amazon EC2 instance. That is, to the Amazon EC2 instance, an Amazon EBS volume appears just like a local disk drive. To write to and read data from Amazon EBS volumes, you use the native file system I/O interfaces of your chosen operating system.

Cost Model

As with other AWS services, with Amazon EBS in most scenarios you pay only for what you use. Amazon EBS pricing has three components: provisioned storage, I/O requests, and snapshot storage. Amazon EBS Magnetic volumes are charged per GB-month of provisioned storage and per million I/O requests. Amazon EBS General Purpose (SSD) volumes are charged per GB-month of provisioned storage, and Amazon EBS Provisioned IOPS (SSD) volumes are charged per GB-month of provisioned storage and per Provisioned IOPS-month. For all volume

types, Amazon EBS snapshots are charged per GB-month of data stored. Amazon EBS snapshot copy is charged for the data transferred between regions, and for the standard Amazon EBS snapshot charges in the destination region.

It's important to remember that for Amazon EBS volumes, you are charged for provisioned (allocated) storage, whether or not you actually use it. For Amazon EBS snapshots, you are charged only for storage actually used (consumed). Note that Amazon EBS snapshots are incremental and compressed, so the storage used in any snapshot is generally much less than the storage consumed for an Amazon EBS volume.

Note that there is no charge for transferring information among the various AWS storage offerings (that is, an Amazon EC2 instance transferring information with Amazon EBS, Amazon S3, Amazon RDS, and so on) as long as the storage offerings are within the same AWS region.

You can find pricing information for Amazon EBS at [the Amazon EBS pricing page](#).²²

Amazon EC2 Instance Storage

[Amazon EC2 instance store volumes](#) (also called ephemeral drives) provide temporary block-level storage for many Amazon EC2 instance types.²³ This storage consists of a preconfigured and pre-attached block of disk storage on the same physical server that hosts the Amazon EC2 instance for which the block provides storage. The amount of the disk storage provided varies by Amazon EC2 instance type. In the Amazon EC2 instance families that provide instance storage, larger instances tend to provide both more and larger instance store volumes.

Note that some instance types, such as the micro instances (t1, t2) and the Compute Optimized C4 instances, use Amazon EBS storage only with no instance storage provided. Note also that instances using Amazon EBS for the root device (in other words, that boot from Amazon EBS) don't expose the instance store volumes by default. You can choose to expose the instance store volumes at

instance launch time by specifying a block device mapping. For more information, see [Block Device Mapping](#) in the *Amazon EC2 User Guide*.²⁴

In addition, the storage-optimized (I2) and dense-storage (D2) Amazon EC2 instance families provide special-purpose instance storage targeted to specific use cases. I2 instances provide very fast SSD-backed instance storage capable of supporting over 365,000 random read IOPS and 315,000 first write IOPS, and I2 instances are optimized for very high random I/O performance and low cost per IOPS. In contrast, D2 instances are optimized for very high storage density, low storage cost, and high sequential I/O performance, offering up to 3.5 GB/second read performance and 3.1 GB/second write performance with a 2 MB block size and up to 48 terabytes of HDD-based local storage.

Usage Patterns

In general, EC2 local instance store volumes are ideal for temporary storage of information that is continually changing, such as buffers, caches, scratch data, and other temporary content, or for data that is replicated across a fleet of instances, such as a load-balanced pool of web servers. Amazon EC2 instance storage is well-suited for this purpose. It consists of the virtual machine's boot device (for instance store AMIs only), plus one or more additional volumes that are dedicated to the Amazon EC2 instance (for both Amazon EBS AMIs and instance store AMIs). This storage can only be used from a single Amazon EC2 instance during that instance's lifetime. Note that, unlike Amazon EBS volumes, instance store volumes cannot be detached or attached to another instance.

For high I/O and high storage, use Amazon EC2 instance storage targeted to these use cases. High I/O instances (the I2 family) provide instance store volumes backed by SSD, and are ideally suited for many high-performance database workloads. Example applications include NoSQL databases like Cassandra and MongoDB, clustered databases, online transaction processing (OLTP) systems, and similar. High storage instances (the D2 family) support much higher storage density per Amazon EC2 instance, and are ideally suited for applications that benefit from high sequential I/O performance across very large datasets. Example applications include data warehouses, Hadoop/MapReduce storage nodes, parallel file systems, and similar.

Note that applications using instance storage for persistent data generally provide data durability through replication, or by periodically copying data to durable storage.

Amazon EC2 instance store volumes don't suit all storage situations. Listed following are some storage needs for which you should consider other AWS storage options instead of Amazon EC2 instance store volumes.

Storage Need	Solution	AWS Services
Persistent storage	If you need persistent virtual disk storage similar to a physical disk drive for files or other data that must persist longer than the lifetime of a single Amazon EC2 instance, Amazon EBS volumes or Amazon S3 are more appropriate.	Amazon EC2 Amazon EBS Amazon S3
Relational database storage	In most cases, relational databases require storage that persists beyond the lifetime of a single Amazon EC2 instance, making Amazon EBS volumes the natural choice.	Amazon EC2 Amazon EBS
Shared storage	Instance store volumes are dedicated to a single Amazon EC2 instance and cannot be shared with other systems or users. If you need storage that can be detached from one instance and attached to a different instance, or if you need the ability to share data easily, Amazon EFS, Amazon S3, or Amazon EBS volumes are the better choice.	Amazon EFS Amazon S3 Amazon EBS
Snapshots	If you need the convenience, long-term durability, availability, and the ability to share point-in-time disk snapshots, Amazon EBS volumes are a better choice.	Amazon EBS

Performance

The instance store volumes that are not SSD-based in most Amazon EC2 instance families have performance characteristics similar to standard Amazon EBS volumes. Because the Amazon EC2 instance virtual machine and the local instance store volumes are located on the same physical server, interaction with this storage is very fast, particularly for sequential access. To increase aggregate IOPS, or to improve sequential disk throughput, multiple instance store volumes can be grouped together using RAID 0 (disk striping) software. Because the bandwidth of the disks is not limited by the network, aggregate sequential throughput for multiple instance volumes can be higher than for the same number of Amazon EBS volumes.

Because of the way that Amazon EC2 virtualizes disks, the first write operation to any location on an instance store volume performs more slowly than subsequent writes. For most applications, amortizing this cost over the lifetime of the instance is acceptable. However, if you require high disk performance, we recommend that you prewarm your drives by writing once to every drive location before production use. The I2, R3, and HI1 instance types use direct-attached SSD backing that provides maximum performance at launch time, without prewarming.

Additionally, R3 and I2 instance store-backed volumes support the TRIM command on Linux instances. For these volumes, you can use TRIM to notify the SSD controller whenever you no longer need data that you've written. This notification lets the controller free space, which can reduce write amplification and increase performance.

The SSD instance store volumes in Amazon EC2 high I/O instances provide from tens of thousands to hundreds of thousands of low-latency, random 4 KB random IOPS. Because of the I/O characteristics of SSD devices, write performance can be variable. For more information, see [High I/O Instances](#) in the *Amazon EC2 User Guide*.²⁵

The instance store volumes in Amazon EC2 high storage instances provide very high storage density and high sequential read and write performance. For more information, see [High Storage Instances](#) in the *Amazon EC2 User Guide*.²⁶

Durability and Availability

Amazon EC2 local instance store volumes are not intended to be used as durable disk storage. Unlike Amazon EBS volume data, data on instance store volumes persists only during the life of the associated Amazon EC2 instance. This functionality means that data on instance store volumes is persistent across orderly instance reboots, but if the Amazon EC2 instance is stopped and restarted, terminates, or fails, all data on the instance store volumes is lost. For more information on the lifecycle of an Amazon EC2 instance, see [Instance Lifecycle](#) in the *Amazon EC2 User Guide*.²⁷

You should not use local instance store volumes for any data that must persist over time, such as permanent file or database storage, without providing data persistence by replicating data or periodically copying data to durable storage

such as Amazon EBS or Amazon S3. Note that this usage recommendation also applies to the special-purpose SSD and high-density instance store volumes in the high I/O and high storage instance types.

Scalability and Elasticity

The number and storage capacity of Amazon EC2 local instance store volumes are fixed and defined by the instance type. Although you can't increase or decrease the number of instance store volumes on a single Amazon EC2 instance, this storage is still scalable and elastic; you can scale the total amount of instance store up or down by increasing or decreasing the number of running Amazon EC2 instances.

To achieve full storage elasticity, include one of the other suitable storage options, such as Amazon S3 or Amazon EFS, in your Amazon EC2 storage strategy.

Security

IAM helps you securely control which users can perform operations such as launch and termination of EC2 instances in your account, and instance store volumes can only be mounted and accessed by the EC2 instances they belong to. Also, when you stop or terminate an instance, the applications and data in its instance store are erased, so no other instance can have access to the instance store in the future.

Access to an EC2 instance is controlled by the guest operating system. If you are concerned about the privacy of sensitive data stored in an instance storage volume, we recommend encrypting your data for extra protection. You can do so by using your own encryption tools, or by using third-party encryption tools available on the [AWS Marketplace](#).²⁸

Interfaces

There is no separate management API for Amazon EC2 instance store volumes. Instead, instance store volumes are specified using the block device mapping feature of the Amazon EC2 API and the AWS Management Console. You cannot create or destroy instance store volumes, but you can control whether or not they are exposed to the Amazon EC2 instance, and what device name is mapped to for each volume.

There is also no separate data API for instance store volumes. Just like Amazon EBS volumes, instance store volumes present a block-device interface to the Amazon EC2 instance. That is, to the Amazon EC2 instance, an instance store volume appears just like a local disk drive. To write to and read data from instance store volumes, you use the native file system I/O interfaces of your chosen operating system.

Note that in some cases, a local instance store volume device is attached to an Amazon EC2 instance upon launch, but must be formatted with an appropriate file system and mounted before use. Also, keep careful track of your block device mappings. There is no simple way for an application running on an Amazon EC2 instance to determine which block device is an instance store (ephemeral) volume and which is an Amazon EBS (persistent) volume.

Cost Model

The cost of an Amazon EC2 instance includes any local instance store volumes, if the instance type provides them. Although there is no additional charge for data storage on local instance store volumes, note that data transferred to and from Amazon EC2 instance store volumes from other Availability Zones or outside of an Amazon EC2 region can incur data transfer charges, and additional charges apply for use of any persistent storage, such as Amazon S3, Amazon Glacier, Amazon EBS volumes, and Amazon EBS snapshots. You can find pricing information for Amazon EC2, Amazon EBS, and data transfer at [the Amazon EC2 pricing page](#).²⁹

AWS Storage Gateway

[The AWS Storage Gateway service](#) connects an on-premises software appliance with cloud-based storage to provide seamless and secure integration between an organization's on-premises IT environment and the AWS storage infrastructure. The service enables you to securely store data in the AWS cloud for scalable and cost-effective storage. AWS Storage Gateway supports industry-standard storage protocols that work with your existing applications. It provides low-latency performance by maintaining frequently accessed data on-premises while securely storing all of your data encrypted in Amazon S3 or Amazon Glacier. For disaster

recovery scenarios, AWS Storage Gateway, together with Amazon EC2, can serve as a cloud-hosted solution that mirrors your entire production environment.

You can download the AWS Storage Gateway software appliance as a virtual machine (VM) image that you install on a host in your data center or as an EC2 instance. Once you've installed your gateway and associated it with your AWS account through the AWS activation process, you can use the AWS Management Console to create gateway-cached volumes, gateway-stored volumes, or a gateway-virtual tape library (VTL), each of which can be mounted as an iSCSI device by your on-premises applications.

With gateway-cached volumes, you can use Amazon S3 to hold your primary data, while retaining some portion of it locally in a cache for frequently accessed data. Gateway-cached volumes minimize the need to scale your on-premises storage infrastructure, while still providing your applications with low-latency access to their frequently accessed data. You can create storage volumes up to 32 TB in size and mount them as iSCSI devices from your on-premises application servers. Each gateway configured for gateway-cached volumes can support up to 20 volumes and total volume storage of 150 TB. Data written to these volumes is stored in Amazon S3, with only a cache of recently written and recently read data stored locally on your on-premises storage hardware.

Gateway-stored volumes store your primary data locally, while asynchronously backing up that data to AWS. These volumes provide your on-premises applications with low-latency access to their entire datasets, while providing durable, off-site backups. You can create storage volumes up to 1 TB in size and mount them as iSCSI devices from your on-premises application servers. Each gateway configured for gateway-stored volumes can support up to 12 volumes and total volume storage of 12 TB. Data written to your gateway-stored volumes is stored on your on-premises storage hardware, and asynchronously backed up to Amazon S3 in the form of Amazon EBS snapshots.

A gateway-VTL allows you to perform offline data archiving by presenting your existing backup application with an iSCSI-based virtual tape library consisting of a virtual media changer and virtual tape drives. You can create virtual tapes in your VTL by using the AWS Management Console, and you can size each virtual tape from 100 GB to 2.5 TB. A VTL can hold up to 1,500 virtual tapes, with a maximum aggregate capacity of 150 TB. Once the virtual tapes are created, your

backup application can discover them by using its standard media inventory procedure. Once created, tapes are available for immediate access and are stored in Amazon S3.

Virtual tapes that you need to access frequently should be stored in a VTL. Data that you don't need to retrieve frequently can be archived to your virtual tape shelf (VTS), which is stored in Amazon Glacier, further reducing your storage costs.

Usage Patterns

Organizations are using AWS Storage Gateway to support a number of use cases. These use cases include corporate file sharing, enabling existing on-premises backup applications to store primary backups on Amazon S3, disaster recovery, and mirroring data to cloud-based compute resources and then later archiving it to Amazon Glacier.

Performance

Because the AWS Storage Gateway VM sits between your application, Amazon S3, and underlying on-premises storage, the performance you experience depends upon a number of factors. These factors include the speed and configuration of your underlying local disks, the network bandwidth between your iSCSI initiator and gateway VM, the amount of local storage allocated to the gateway VM, and the bandwidth between the gateway VM and Amazon S3. For gateway-cached volumes, to provide low-latency read access to your on-premises applications, it's important that you provide enough local cache storage to store your recently accessed data. The [AWS Storage Gateway documentation](#) provides guidance on how to optimize your environment setup for best performance, including how to properly size your local storage.³⁰

AWS Storage Gateway efficiently uses your Internet bandwidth to speed up the upload of your on-premises application data to AWS. AWS Storage Gateway only uploads data that has changed, which minimizes the amount of data sent over the Internet. To further increase throughput and reduce your network costs, you can also use [AWS Direct Connect](#) to establish a dedicated network connection between your on-premises gateway and AWS.³¹

Durability and Availability

AWS Storage Gateway durably stores your on-premises application data by uploading it to Amazon S3 or Amazon Glacier. Both of these AWS services store data in multiple facilities and on multiple devices within each facility, being designed to provide an average annual durability of 99.999999999 percent. They also perform regular, systematic data integrity checks and are built to be automatically self-healing.

Scalability and Elasticity

In both gateway-cached and gateway-stored volume configurations, AWS Storage Gateway stores data in Amazon S3, which has been designed to offer a very high level of scalability and elasticity automatically. Unlike a typical file system that can encounter issues when storing large number of files in a directory, Amazon S3 supports a virtually unlimited number of files in any bucket. Also, unlike a disk drive that has a limit on the total amount of data that can be stored before you must partition the data across drives or servers, an Amazon S3 bucket can store a virtually unlimited number of bytes. You are able to store any number of objects, and Amazon S3 will manage scaling and distributing redundant copies of your information onto other servers in other locations in the same region, all using Amazon's high-performance infrastructure.

In a gateway-VTL configuration, AWS Storage Gateway stores data in Amazon S3 or Amazon Glacier, providing a virtual tape infrastructure that scales seamlessly with your business needs and eliminates the operational burden of provisioning, scaling, and maintaining a physical tape infrastructure.

Security

IAM helps you provide security in controlling access to AWS Storage Gateway. With IAM, you can create multiple IAM users under your AWS account. The [AWS Storage Gateway API](#) enables a list of actions each IAM user can perform on AWS Storage Gateway.³²

The AWS Storage Gateway encrypts all data in transit to and from AWS by using SSL. All volume and snapshot data stored in AWS using gateway-stored or gateway-cached volumes and all virtual tape data stored in AWS using a gateway-VTL is encrypted at rest using AES-256, a secure symmetric-key encryption standard using 256-bit encryption keys. AWS Storage Gateway supports

authentication between your gateway and iSCSI initiators by using Challenge-Handshake Authentication Protocol (CHAP).

Interfaces

The AWS Management Console can be used to download the AWS Storage Gateway VM on-premises or onto an Amazon EC2 instance (an AMI that contains the gateway VM image). You can then select between a gateway-cached, gateway-stored, or gateway-VTL configuration and activate your storage gateway by associating your gateway's IP address with your AWS account. All the detailed steps for AWS Storage Gateway deployment can be found in [Getting Started](#) in the *AWS Storage Gateway User Guide*.³³

The integrated AWS CLI also provides a set of high-level, Linux-like commands for common operations of the AWS Storage Gateway service.

You can also use the AWS SDKs to develop applications that interact with AWS Storage Gateway. The AWS SDKs for Java, .NET, JavaScript/Node.js, Ruby, PHP, and Go wrap the underlying AWS Storage Gateway API to simplify your programming tasks.

Cost Model

With AWS Storage Gateway, you pay only for what you use. AWS Storage Gateway has the following pricing components: gateway usage (per gateway per month), snapshot storage usage (per GB per month), volume storage usage (per GB per month), virtual tape shelf storage (per GB per month), virtual tape library storage (per GB per month), retrieval from virtual tape shelf (per GB), and data transfer out (per GB per month). You can find pricing information at [the AWS Storage Gateway pricing page](#).³⁴

AWS Import/Export Snowball

[AWS Import/Export Snowball](#) (AWS Snowball) accelerates moving large amounts of data into and out of AWS using secure Snowball appliances. The AWS Snowball appliance is purpose-built for efficient data storage and transfer. It is rugged enough to withstand a 6 G jolt, and at less than 50 pounds, light enough

for one person to carry. It is entirely self-contained, with power and a 10 GB network connection on the back and an E Ink display and control panel on the front. Each Snowball is weather-resistant and serves as its own shipping container.

AWS transfers your data directly onto and off of Snowball storage devices using Amazon's high-speed internal network and bypassing the Internet. For datasets of significant size, AWS Import/Export Snowball is often faster than Internet transfer and more cost-effective than upgrading your connectivity. AWS Import/Export Snowball supports importing data into and exporting data from Amazon S3 buckets. From there, the data can be copied or moved to other AWS services such as Amazon EBS and Amazon Glacier as desired.

Usage Patterns

AWS Import/Export Snowball is ideal for transferring anywhere from terabytes to many petabytes of data in and out of the AWS cloud securely, especially in cases where you don't want to make expensive upgrades to your network infrastructure, frequently experience large backlogs of data, are in a physically isolated environment, or are in an area where high-speed Internet connections are not available or cost-prohibitive. In general, if loading your data over the Internet would take a week or more, you should consider using AWS Import/Export Snowball.

Common use cases include cloud migration, disaster recovery, data center decommission, and content distribution. When you decommission a data center, many steps are involved to make sure valuable data is not lost, and Snowball can help ensure data is securely and cost-effectively transferred to AWS. In a content distribution scenario, you might use Snowball appliances if you regularly receive or need to share large amounts of data with clients, customers, or business associates. Snowball appliances can be sent directly from AWS to client or customer locations.

However, AWS Import/Export Snowball might not be the ideal solution if your data can be transferred over the Internet in less than one week.

Performance

The Snowball appliance is purpose-built for efficient data storage and transfer, including a high-speed, 10 Gbps network connection designed to minimize data transfer times, allowing you to transfer up to 50 terabytes of data from your data source to the appliance in less than one day, plus shipping time.

You can use the [Snowball client](#) to estimate the time it takes to transfer your data (refer to the *AWS Import/Export User Guide* for more details).³⁵ It can take as little as 13 hours to copy 50 TB of data from your data source to a Snowball appliance using a 10 Gbps local connection, assuming 80 percent network utilization. In this case, the end-to-end time to transfer the data into AWS is approximately a week, including default shipping and handling time in AWS data centers. Copying 100 TB of data can be completed in same amount of time by using two 50 TB Snowballs in parallel.

In general, you can improve your transfer speed from your data source to the Snowball by reducing local network use, eliminating unnecessary hops between the Snowball and the workstation, and using a powerful computer as your workstation.

Durability and Availability

Once the data is imported to AWS, the durability and availability characteristics of the target storage applies. Amazon S3 is designed for 99.999999999 percent (11 nines) durability.

Scalability and Elasticity

Each AWS Import/Export Snowball appliance is capable of storing 50 TB of data. If you want to transfer more data than that, you can use multiple appliances. For Amazon S3, individual files are loaded as objects and can range up to 5 TB in size, but you can load any number of objects in Amazon S3. The aggregate total amount of data that can be imported is virtually unlimited.

Security

[You can integrate AWS Import/Export Snowball with IAM](#) to control which actions a user can perform.³⁶ You can give the IAM users on your AWS account access to all Snowball actions or to a subset of them. Similarly, an IAM user that creates an AWS Import/Export Snowball job must have permissions to access the

Amazon S3 buckets that will be used for the import operations. You can control access to the Snowball by careful distribution of a job's manifest and unlock code.

For AWS Import/Export Snowball, [AWS Key Management Service \(AWS KMS\)](#) protects the encryption keys used to protect data on each Snowball.³⁷ When you create your job, you also choose or create for the job an Amazon Resource Name (ARN), for an AWS KMS key that you own. Specifying the AWS KMS key ARN tells Snowball which AWS KMS master key to use to encrypt the unique keys on the Snowball.

AWS Import/Export Snowball is physically secured by using an industry-standard Trusted Platform Module (TPM) that uses a dedicated processor designed to detect any unauthorized modifications to the hardware, firmware, or software.

Interfaces

The following is an overview of the steps by which a Snowball is implemented to import your data into AWS.

1. To start, you sign in to the AWS Import/Export Snowball Management Console and create a job.
2. AWS then prepares a Snowball for your job.
3. The Snowball is shipped to you through UPS. You can find your tracking number and a link to the UPS tracking website on the AWS Import/Export Snowball Management Console.
4. A few days later, UPS delivers the Snowball to the address you provided when you created the job.
5. Next, you get ready to transfer data by getting your credentials, your job manifest, and the manifest's unlock code, and by downloading the Snowball client. The Snowball client is the tool that you'll use to manage the flow of data from your on-premises data source to the Snowball.
6. Install the Snowball client on the computer workstation that has your data source mounted on it.
7. Move the Snowball into your data center, open it, and connect it to power and your local network.
8. Power on the Snowball and start the Snowball client. You provide the IP address of the Snowball, the path to your manifest, and the unlock code.

The Snowball client decrypts the manifest and uses it to authenticate your access to the Snowball.

9. You then use the Snowball client to transfer the data that you want to import into Amazon S3 from your data source into the Snowball.
10. After your data transfer is complete, you power off the Snowball and unplug its cables. The E Ink shipping label automatically updates to show the correct AWS facility to ship to, and you can track job status by using [Amazon Simple Notification Services \(Amazon SNS\)](#), text messages, or directly in the console.³⁸
11. UPS returns the Snowball to AWS.
12. AWS gets the Snowball and imports your data into Amazon S3. On average, it takes about a day for AWS to begin importing your data into S3, and the import can take a few days. If there are any complications or issues, we contact you through email.

Once the data transfer job has been processed and verified, AWS performs a software erasure of the Snowball appliance that follows the National Institute of Standards and Technology (NIST) guidelines for media sanitization.

Cost Model

With AWS Import/Export Snowball, as with most other AWS services, you pay only for what you use. AWS Import/Export Snowball has three pricing components: service fee (per job), extra day charges as required (the first 10 days of onsite usage are free), and data transfer. For the destination storage, the standard Amazon S3 storage pricing applies. You can find pricing information at [the AWS Import/Export pricing page](#).³⁹



Amazon CloudFront

[Amazon CloudFront](#) is a content-delivery web service that speeds up the distribution of your website's dynamic, static, and streaming content by making it available from a global network of edge locations. When a user requests content that you're serving with Amazon CloudFront, the user is routed to the edge location that provides the lowest latency (time delay), so content is delivered with better performance than if the user had accessed the content from a data center

farther away. If the content is already in the edge location with the lowest latency, Amazon CloudFront delivers it immediately. If the content is not currently in that edge location, Amazon CloudFront retrieves it from an Amazon S3 bucket or an HTTP server (for example, a web server) that you have identified as the source for the definitive version of your content. Amazon CloudFront caches content at edge locations for a period of time that you specify.

Amazon CloudFront supports all files that can be served over HTTP. These files include dynamic web pages, such as HTML or PHP pages, and any popular static files that are a part of your web application, such as website images, audio, video, media files or software downloads. For on-demand media files, you can also choose to stream your content using Real-Time Messaging Protocol (RTMP) delivery. Amazon CloudFront also supports delivery of live media over HTTP.

Amazon CloudFront is optimized to work with other Amazon web services, such as Amazon S3, Amazon EC2, Elastic Load Balancing, and Amazon Route 53. Amazon CloudFront also works seamlessly with any non-AWS origin servers that store the original, definitive versions of your files.

Usage Patterns

Amazon CloudFront is ideal for distribution of frequently accessed static content that benefits from edge delivery, such as popular website images, videos, media files or software downloads. Amazon CloudFront can also be used to deliver dynamic web applications over HTTP. These applications can include static content, dynamic content, or a whole site with a mixture of the two. Amazon CloudFront is also commonly used to stream audio and video files to web browsers and mobile devices. To get a better understanding of your end user usage patterns, you can use [Amazon CloudFront reports](#).⁴⁰

If you need to remove an object from Amazon CloudFront edge-server caches before it expires, you can either [invalidate the object](#) or use [object versioning](#) to serve a different version of the object that has a different name.⁴¹ ⁴² Additionally, it might be better to serve infrequently accessed data directly from the origin server, avoiding the additional cost of origin fetches for data that is not likely to be reused at the edge.

Performance

Amazon CloudFront is designed for low-latency and high-bandwidth delivery of content. Amazon CloudFront speeds up the distribution of your content by routing end users to the edge location that can best serve each end user's request in a worldwide network of edge locations. Typically, requests are routed to the nearest Amazon CloudFront edge location in terms of latency. This approach dramatically reduces the number of networks that your users' requests must pass through and improves performance. Users get both lower latency—here latency is the time it takes to load the first byte of an object—and the higher sustained data transfer rates needed to deliver popular objects at scale.

Durability and Availability

Because a CDN is an edge cache, Amazon CloudFront does not provide durable storage. The origin server, such as Amazon S3 or a web server running on Amazon EC2, provides the durable file storage needed. Amazon CloudFront provides high availability by using a distributed global network of edge locations. Origin requests from the edge locations to AWS origin servers (for example, Amazon EC2, Amazon S3, and so on) are carried over network paths that Amazon constantly monitors and optimizes for both availability and performance. This edge network provides increased reliability and availability because there is no longer a central point of failure. Copies of your files are now held in edge locations around the world.

Scalability and Elasticity

Amazon CloudFront is designed to provide seamless scalability and elasticity. You can easily start very small and grow to massive numbers of global connections. With Amazon CloudFront, you don't need to worry about maintaining expensive web server capacity to meet the demand from potential traffic spikes for your content. The service automatically responds as demand spikes and fluctuates for your content, without any intervention from you. Amazon CloudFront also uses multiple layers of caching at each edge location and collapses simultaneous requests for the same object before contacting your origin server. These optimizations further reduce the need to scale your origin infrastructure as your website becomes more popular.

Security

Amazon CloudFront is a very secure service to distribute your data. It integrates with IAM so that you can create users for your AWS account and specify which Amazon CloudFront actions a user (or a group of users) can perform in your AWS account.

You can configure Amazon CloudFront to create log files that contain detailed information about every user request that Amazon CloudFront receives. These [access logs](#) are available for both web and RTMP distributions.⁴³ Additionally, Amazon CloudFront [integrates with Amazon CloudWatch metrics](#) so that you can monitor your website or application.⁴⁴

Interfaces

You can manage and configure Amazon CloudFront in several ways. The AWS Management Console provides an easy way to manage Amazon CloudFront and supports all features of the Amazon CloudFront API. For example, you can enable or disable distributions, configure CNAMEs, and enable end-user logging using the console. You can also use the Amazon CloudFront command line tools, the native REST API, or one of the supported SDKs.

There is no data API for Amazon CloudFront, and no command to preload data. Instead, data is automatically pulled into Amazon CloudFront edge locations on the first access of an object from that location.

Cost Model

With Amazon CloudFront, there are no long-term contracts or required minimum monthly commitments—you pay only for as much content as you actually deliver through the service. Amazon CloudFront has two pricing components: regional data transfer out (per GB) and requests (per 10,000). As part of the Free Usage Tier, new AWS customers don't get charged for 50 GB data transfer out and 2,000,000 HTTP and HTTPS requests each month for one year.

Note that if you use an AWS service as the origin (for example, Amazon S3, Amazon EC2, Elastic Load Balancing, or others), data transferred from origin to edge locations (Amazon CloudFront “origin fetches”) will be free of charge. Data transfer out of Amazon CloudFront to your origin server will be billed at the “Regional Data Transfer Out of Origin” rates.

CloudFront provides three different price classes according to where your content needs to be distributed. If you don't need your content to be distributed globally, but only within certain locations such as the US and Europe, you can lower the prices you pay to deliver by choosing a price class that includes only these locations.

Although there are no long-term contracts or required minimum monthly commitments, CloudFront offers an optional reserved capacity plan that gives you the option to commit to a minimum monthly usage level for 12 months or longer and in turn receive a significant discount. You can find pricing information at [the Amazon CloudFront pricing page](#).⁴⁵

Contributors

The following individuals contributed to this document:

- Shruti Worlikar, Solutions Architect, Amazon Web Services
- Fabio Silva, Solutions Architect, Amazon Web Services

References and Further Reading

AWS Storage Services

[Amazon S3](#)⁴⁶

[Amazon Glacier](#)⁴⁷

[Amazon EBS](#)⁴⁸

[Amazon EC2 Instance Store](#)⁴⁹

[AWS Storage Gateway](#)⁵⁰

[AWS Import/Export Snowball](#)⁵¹

[Amazon CloudFront](#)⁵²

Other Resources

[AWS SDKs, IDE Toolkits, and Command Line Tools](#)⁵³

[Amazon Web Services Simple Monthly Calculator](#)⁵⁴

[Amazon Web Services Blog](#)⁵⁵

[Amazon Web Services Forums](#)⁵⁶

[AWS Free Usage Tier](#)⁵⁷

[AWS Case Studies](#)⁵⁸

Notes

¹ <https://docs.aws.amazon.com/AmazonS3/latest/dev/crr.html>

² <http://docs.aws.amazon.com/AmazonS3/latest/dev/uploadobjusingmpu.html>

³ <http://docs.aws.amazon.com/AmazonS3/latest/dev/access-control-overview.html#access-control-resources-manage-permissions-basics>

⁴ <http://docs.aws.amazon.com/AmazonS3/latest/dev/serv-side-encryption.html>

⁵ <http://docs.aws.amazon.com/AmazonS3/latest/dev/UsingClientSideEncryption.html>

⁶

<http://docs.aws.amazon.com/AmazonS3/latest/dev/Versioning.html#MultiFactorAuthenticationDelete>

⁷ <http://docs.aws.amazon.com/AmazonS3/latest/dev/ServerLogs.html>

⁸ <http://aws.amazon.com/sns/>

⁹ <http://aws.amazon.com/sqs/>

¹⁰ <http://aws.amazon.com/lambda/>

¹¹ <http://aws.amazon.com/free/>

¹² <http://aws.amazon.com/s3/pricing/>

¹³ <http://docs.aws.amazon.com/amazonglacier/latest/dev/uploading-archive-mpu.html>

¹⁴ <http://docs.aws.amazon.com/amazonglacier/latest/dev/downloading-an-archive.html#downloading-an-archive-range>

¹⁵ <https://aws.amazon.com/iam>

- 16 <http://aws.amazon.com/cloudtrail/>
- 17 <http://aws.amazon.com/sns/>
- 18 <http://docs.aws.amazon.com/AmazonS3/latest/dev/object-lifecycle-mgmt.html>
- 19 <http://aws.amazon.com/glacier/pricing/>
- 20 <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-initialize.html>
- 21 <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSEncryption.html>
- 22 <http://aws.amazon.com/ebs/pricing/>
- 23 <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/InstanceStorage.html>
- 24 <http://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/block-device-mapping-concepts.html>
- 25 <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/i2-instances.html>
- 26 http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/high_storage_instances.html
- 27 <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-lifecycle.html>
- 28 <https://aws.amazon.com/marketplace>
- 29 <http://aws.amazon.com/ec2/pricing/>
- 30 <http://docs.aws.amazon.com/storagegateway/latest/userguide/WhatIsStorageGateway.html>
- 31 <http://aws.amazon.com/directconnect/>
- 32 <http://docs.aws.amazon.com/storagegateway/latest/userguide/AWSStorageGatewayAPI.html>
- 33 <http://docs.aws.amazon.com/storagegateway/latest/userguide/GettingStarted-common.html>
- 34 <http://aws.amazon.com/storagegateway/pricing/>
- 35 <http://aws.amazon.com/importexport/tools>
- 36 <http://docs.aws.amazon.com/AWSImportExport/latest/DG/auth-access-control.html>
- 37 <https://aws.amazon.com/kms/>
- 38 <https://aws.amazon.com/sns/>
- 39 <http://aws.amazon.com/importexport/pricing/>
- 40 <http://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/reports.html>
- 41 <http://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/Invalidation.html>

42

<http://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/ReplacingObjects.html>

43 <http://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/AccessLogs.html>44 <http://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/monitoring-using-cloudwatch.html>45 <http://aws.amazon.com/cloudfront/pricing/>46 <http://aws.amazon.com/s3/>47 <http://aws.amazon.com/glacier/>48 <http://aws.amazon.com/ebs/>49 <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/InstanceStorage.html>50 <http://aws.amazon.com/storagegateway/>51 <http://aws.amazon.com/importexport/>52 <http://aws.amazon.com/cloudfront/>53 <http://aws.amazon.com/tools/>54 <http://calculator.s3.amazonaws.com/index.html>55 <https://aws.amazon.com/blogs/aws/>56 <https://forums.aws.amazon.com/index.jspa>57 <http://aws.amazon.com/free/>58 <http://aws.amazon.com/solutions/case-studies/>