



# AWS STORAGE SERVICES

**KiranReddy Adulla**

<https://www.linkedin.com/in/kiranreddy-adulla/>

## Contents

|  |          |
|--|----------|
| <b>Instance Stores and Amazon Elastic Block Store (Amazon EBS) .....</b> | <b>3</b> |
| Instance stores.....   | 3        |
| Amazon Elastic Block Storage (Amazon EBS).....                           | 6        |
| <b>Amazon EBS Snapshots .....</b>  | <b>6</b> |
| <b>Amazon Simple Storage Service (Amazon S3).....</b>                    | <b>7</b> |
| Object Storage .....   | 7        |

|  |           |
|--|-----------|
| <b>Amazon Simple Storage Service (Amazon S3)</b>       | <b>7</b>  |
| <b>Amazon S3 Storage Classes</b>                       | <b>8</b>  |
| S3 Standard  | 8         |
| S3 Standard-Infrequent Access (S3 Standard-IA)         | 8         |
| S3 One Zone-Infrequent Access (S3 One Zone-IA)         | 9         |
| S3 Intelligent-Tiering                                 | 9         |
| S3 Glacier   | 9         |
| S3 Glacier Deep Archive                                | 9         |
| <b>Amazon EBS vs Amazon S3</b>                         | <b>10</b> |
| <b>Amazon EBS and Amazon S3</b>                        | <b>10</b> |
| <b>Amazon EFS</b>                                      | <b>12</b> |
| <b>Amazon Elastic File System (Amazon EFS)</b>         | <b>13</b> |
| File Storage   | 13        |
| Comparing Amazon EBS and Amazon EFS                    | 13        |
| <b>Amazon Relational Database Service (Amazon RDS)</b> | <b>14</b> |
| Relational databases                                   | 14        |
| Amazon Relational Database Service                     | 14        |
| <b>Amazon RDS database engines</b>                     | <b>15</b> |
| Amazon Aurora  | 15        |
| Amazon DynamoDB  | 15        |
| Nonrelational Databases                                | 16        |
| Amazon DynamoDB  | 16        |
| Serverless   | 16        |
| Automatic Scaling                                      | 16        |
| <b>AMAZON RDS vs AMAZON DYNAMO DB</b>                  | <b>17</b> |
| <b>Amazon Redshift</b>                                 | <b>19</b> |
| <b>AWS DMS (Data Migration Serves)</b>                 | <b>19</b> |
| Homogeneous migrations                                 | 19        |
| heterogeneous migration                                | 19        |
| <b>Additional Database Services</b>                    | <b>19</b> |
| Amazon DocumentDB                                      | 20        |
| Amazon Neptune   | 20        |
| Amazon Quantum Ledger Database (Amazon QLDB)           | 20        |

|                                   |    |
|-----------------------------------|----|
| Amazon Managed Blockchain.....    | 20 |
| Amazon ElastiCache.....           | 20 |
| Amazon DynamoDB Accelerator ..... | 20 |

## Instance Stores and Amazon Elastic Block Store (Amazon EBS)

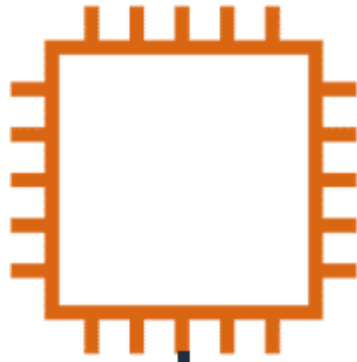
### Instance stores

Block-level storage volumes behave like physical hard drives.

An [instance store](#) provides temporary block-level storage for an Amazon EC2 instance. An instance store is disk storage that is physically attached to the host computer for an EC2 instance, and therefore has the same lifespan as the instance. When the instance is terminated, you lose any data in the instance store.

**An Amazon EC2 instance with an attached instance store is running.**

Amazon EC2  
instance

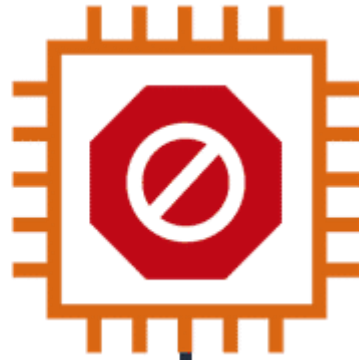


Instance store  
with data



The instance is stopped or terminated.

Amazon EC2  
instance

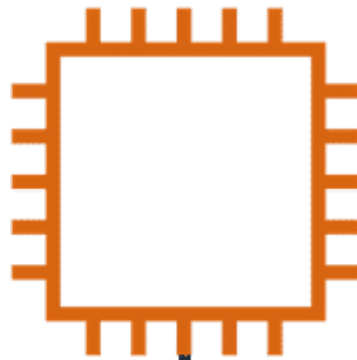


Instance store  
with data



All data on the attached instance store is deleted.

Amazon EC2  
instance



Instance store  
without data

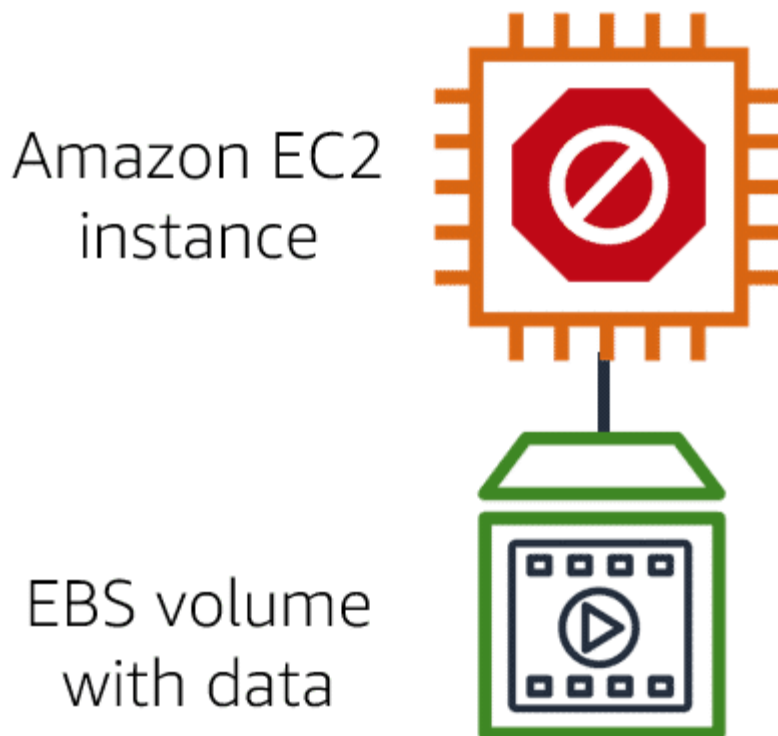


## Amazon Elastic Block Storage (Amazon EBS)

[Amazon Elastic Block Store \(Amazon EBS\)](#) is a service that provides block-level storage volumes that you can use with Amazon EC2 instances. If you stop or terminate an Amazon EC2 instance, all the data on the attached EBS volume remains available.

To create an EBS volume, you define the configuration (such as volume size and type) and provision it. After you create an EBS volume, it can attach to an Amazon EC2 instance.

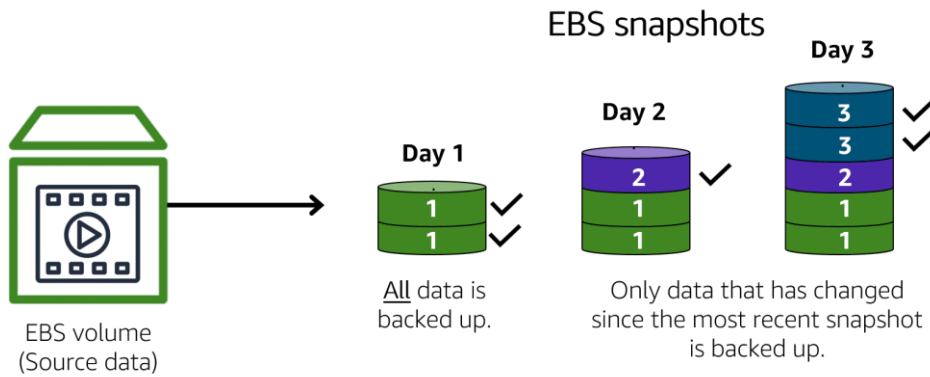
Because EBS volumes are for data that needs to persist, it's important to back up the data. You can take incremental backups of EBS volumes by creating Amazon EBS snapshots.



## Amazon EBS Snapshots

An [EBS snapshot](#) is an incremental backup. This means that the first backup taken of a volume copies all the data. For subsequent backups, only the blocks of data that have changed since the most recent snapshot are saved.

Incremental backups are different from full backups, in which all the data in a storage volume copies each time a backup occurs. The full backup includes data that has not changed since the most recent backup.

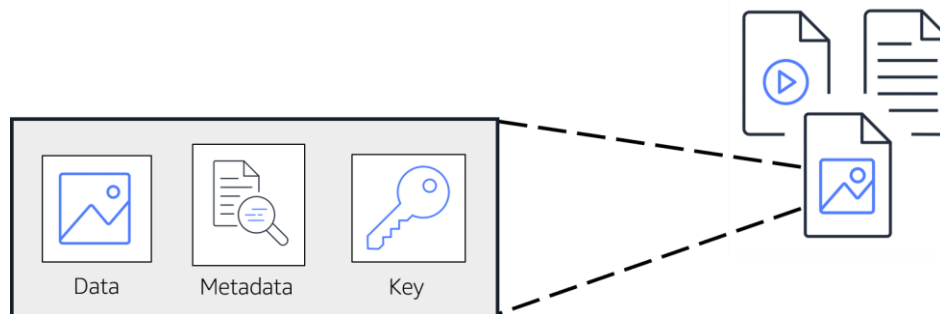


## Amazon Simple Storage Service (Amazon S3)

### Object Storage

In **object storage**, each object consists of data, metadata, and a key.

The data might be an image, video, text document, or any other type of file. Metadata contains information about what the data is, how it is used, the object size, and so on. An object's key is its unique identifier.



Recall that when you modify a file in block storage, only the pieces that are changed are updated. When a file in object storage is modified, the entire object is updated.

## Amazon Simple Storage Service (Amazon S3)

[Amazon Simple Storage Service \(Amazon S3\)](#) is a service that provides object-level storage. Amazon S3 stores data as objects in buckets.

You can upload any type of file to Amazon S3, such as images, videos, text files, and so on. For example, you might use Amazon S3 to store backup files, media files for a website, or archived documents. Amazon S3 offers unlimited storage space. The maximum file size for an object in Amazon S3 is 5 TB.

When you upload a file to Amazon S3, you can set permissions to control visibility and access to it. You can also use the Amazon S3 versioning feature to track changes to your objects over time.

## Amazon S3 Storage Classes

With Amazon S3, you pay only for what you use. You can choose from [a range of storage classes](#) to select a fit for your business and cost needs. When selecting an Amazon S3 storage class, consider these two factors:

- How often you plan to retrieve your data
- How available you need your data to be

### S3 Standard

- Designed for frequently accessed data
- Stores data in a minimum of three Availability Zones

S3 Standard provides high availability for objects. This makes it a good choice for a wide range of use cases, such as websites, content distribution, and data analytics. S3 Standard has a higher cost than other storage classes intended for infrequently accessed data and archival storage.

### S3 Standard-Infrequent Access (S3 Standard-IA)

- Ideal for infrequently accessed data
- Similar to S3 Standard but has a lower storage price and higher retrieval price

S3 Standard-IA is ideal for data infrequently accessed but requires high availability when needed. Both S3 Standard and S3 Standard-IA store data in a minimum of three Availability Zones. S3 Standard-IA provides the same level of availability as S3 Standard but with a lower storage price and a higher retrieval price.



## **S3 One Zone-Infrequent Access (S3 One Zone-IA)**

- Stores data in a single Availability Zone
- Has a lower storage price than S3 Standard-IA

Compared to S3 Standard and S3 Standard-IA, which store data in a minimum of three Availability Zones, S3 One Zone-IA stores data in a single Availability Zone. This makes it a good storage class to consider if the following conditions apply:

- You want to save costs on storage.
- You can easily reproduce your data in the event of an Availability Zone failure.

## **S3 Intelligent-Tiering**

- Ideal for data with unknown or changing access patterns
- Requires a small monthly monitoring and automation fee per object

In the S3 Intelligent-Tiering storage class, Amazon S3 monitors objects' access patterns. If you haven't accessed an object for 30 consecutive days, Amazon S3 automatically moves it to the infrequent access tier, S3 Standard-IA. If you access an object in the infrequent access tier, Amazon S3 automatically moves it to the frequent access tier, S3 Standard.

## **S3 Glacier**

- Low-cost storage designed for data archiving
- Able to retrieve objects within a few minutes to hours

S3 Glacier is a low-cost storage class that is ideal for data archiving. For example, you might use this storage class to store archived customer records or older photos and video files.

## **S3 Glacier Deep Archive**

- Lowest-cost object storage class ideal for archiving
- Able to retrieve objects within 12 hours

When deciding between Amazon S3 Glacier and Amazon S3 Glacier Deep Archive, consider how quickly you need to retrieve archived objects. You can retrieve objects stored in the S3 Glacier storage class within a few minutes to a few hours. By comparison, you can retrieve objects stored in the S3 Glacier Deep Archive storage class within 12 hours.

## Amazon EBS vs Amazon S3

**EBS**

**Amazon Elastic Block Store**

- Sizes up to 16 TiB
- Survive termination of their EC2 instance
- Solid state by default
- HDD options

The image shows a comparison between Amazon EBS and Amazon S3. On the left, under the heading 'EBS', is a white icon of a cylinder with four arrows pointing outwards from its top and bottom edges. On the right, under the heading 'Amazon Elastic Block Store', is a blue box containing four features, each with a green checkmark icon: 'Sizes up to 16 TiB', 'Survive termination of their EC2 instance', 'Solid state by default', and 'HDD options'. A 'Share' button is visible in the top right corner of the blue box.

**Amazon Simple Storage Service**

- Unlimited storage
- Individual objects up to 5 TBs
- Write once/read many
- 99.999999999% durability

**S3**

The image shows a comparison between Amazon S3 and Amazon EBS. On the left, under the heading 'Amazon Simple Storage Service', is an orange box containing four features, each with a green checkmark icon: 'Unlimited storage', 'Individual objects up to 5 TBs', 'Write once/read many', and '99.999999999% durability'. On the right, under the heading 'S3', is a white icon of a bucket. A 'Share' button is visible in the top right corner of the orange box. At the bottom of the image, the text 'They are Amazon Simple' is partially visible.

## Amazon EBS and Amazon S3

When you store data in EBS, it's break file into components or blocks

When you store data in S3, it stores data as a complete discrete object

**Round 1:**

Let's say you're running a photo analysis website where users upload a photo of themselves and your application finds the animals that look just like them.

You have potentially millions of animal pictures that all need to be indexed and possibly viewed by thousands of people at once.

This is the perfect use case for S3.

S3 is already web-enabled.

Every object already has a URL that you can control access rights to who can see or manage the image.

It's regionally distributed, which means that it has 11 9's of durability.

No need to worry about backup strategies. S3 is your backup strategy. Plus the cost savings is substantial overrunning the same storage load on EBS, with the additional advantage of being server less. No Amazon EC2 instances are needed.

**Round 2:**

you have an 80-gigabyte video file that you're making edit corrections on.

To know the best storage class here, we need to understand the difference between object storage and block storage. Object storage treats any file as a complete discrete object.

Now, this is great for documents, and images, and video files that get uploaded and consumed as entire objects.

But every time there's a change to the object, you must re-upload the entire file.

There are no Delta updates.

Block storage breaks those files down to small component parts or blocks. This means for that 80-gigabyte file, when you make an edit to one scene in the film and save that change, the engine only updates the blocks where those bits live. If you're making a bunch of micro edits using EBS, Elastic Block Storage is the perfect use case.

If you were using S3, every time you save the changes, the system would have to upload all 80 gigabytes, that whole thing every time.

## Amazon EFS

Amazon EFS:  
Multiple instances can access the data in EFS at the same time.

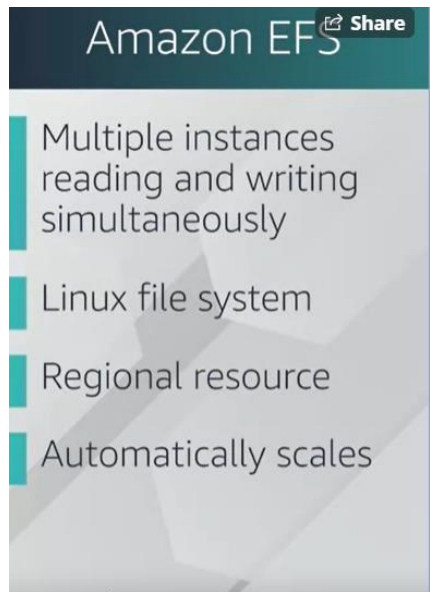
### Amazon EBS Share

- Volumes attach to EC2 instances

- Availability Zone level resource

- Need to be in the same Availability Zone to attach EC2 instances

- Volumes do not automatically scale



## Amazon Elastic File System (Amazon EFS)

### File Storage

In **file storage**, multiple clients (such as users, applications, servers, and so on) can access data that is stored in shared file folders. In this approach, a storage server uses block storage with a local file system to organize files. Clients access data through file paths.

Compared to block storage and object storage, file storage is ideal for use cases in which a large number of services and resources need to access the same data at the same time.

[Amazon Elastic File System \(Amazon EFS\)](#) is a scalable file system used with AWS Cloud services and onpremises resources. As you add and remove files, Amazon EFS grows and shrinks automatically. It can scale on demand to petabytes without disrupting applications.

## Comparing Amazon EBS and Amazon EFS

### Amazon EBS

- An Amazon EBS volume stores data in a **single** Availability Zone.
- To attach an Amazon EC2 instance to an EBS volume, both the Amazon EC2 instance and the EBS volume must reside within the same Availability Zone.

### Amazon EFS

- Amazon EFS is a regional service. It stores data in and across **multiple** Availability Zones.

- The duplicate storage enables you to access data concurrently from all the Availability Zones in the Region where a file system is located. Additionally, on-premises servers can access Amazon EFS using AWS Direct Connect.

## Amazon Relational Database Service (Amazon RDS)

### Relational databases

In a **relational database**, data is stored in a way that relates it to other pieces of data.

An example of a relational database might be the coffee shop's inventory management system. Each record in the database would include data for a single item, such as product name, size, price, and so on.

Relational databases use **structured query language (SQL)** to store and query data. This approach allows data to be stored in an easily understandable, consistent, and scalable way. For example, the coffee shop owners can write a SQL query to identify all the customers whose most frequently purchased drink is a medium latte.

Example of data in a relational database:

| ID | Product Name               | Size   | Price  |
|----|----------------------------|--------|--------|
| 1  | Medium roast ground coffee | 12 oz. | \$5.30 |
| 2  | Dark roast ground coffee   | 20 oz. | \$9.27 |

### Amazon Relational Database Service

[Amazon Relational Database Service \(Amazon RDS\)](#) is a service that enables you to run relational databases in the AWS Cloud.

Amazon RDS is a managed service that automates tasks such as hardware provisioning, database setup, patching, and backups. With these capabilities, you can spend less time completing administrative tasks and more time using data to innovate your applications. You can integrate Amazon RDS with other services to fulfill your business and operational needs, such as using AWS Lambda to query your database from a serverless application.

Amazon RDS provides a number of different security options. Many Amazon RDS database engines offer encryption at rest (protecting data while it is stored) and encryption in transit (protecting data while it is being sent and received).

## Amazon RDS database engines

Amazon RDS is available on six database engines, which optimize for memory, performance, or input/output (I/O). Supported database engines include:

- Amazon Aurora
- PostgreSQL
- MySQL
- MariaDB
- Oracle Database
- Microsoft SQL Server

## Amazon Aurora

[Amazon Aurora](#) is an enterprise-class relational database. It is compatible with MySQL and PostgreSQL relational databases. It is up to five times faster than standard MySQL databases and up to three times faster than standard PostgreSQL databases.

Amazon Aurora helps to reduce your database costs by reducing unnecessary input/output (I/O) operations, while ensuring that your database resources remain reliable and available.

Consider Amazon Aurora if your workloads require high availability. It replicates six copies of your data across three Availability Zones and continuously backs up your data to Amazon S3.

## Amazon DynamoDB

## Nonrelational Databases

In a **nonrelational database**, you create tables. A table is a place where you can store and query data.

Nonrelational databases are sometimes referred to as “NoSQL databases” because they use structures other than rows and columns to organize data. One type of structural approach for nonrelational databases is keyvalue pairs. With key-value pairs, data is organized into items (keys), and items have attributes (values). You can think of attributes as being different features of your data.

In a key-value database, you can add or remove attributes from items in the table at any time. Additionally, not every item in the table has to have the same attributes.

Example of data in a nonrelational database:

| Key | Value  |
|-----|--|
| 1   | <b>Name:</b> John Doe<br><b>Address:</b> 123 Any Street<br><b>Favorite drink:</b> Medium latte |
| 2   | <b>Name:</b> Mary Major<br><b>Address:</b> 100 Main Street<br><b>Birthday:</b> July 5, 1994    |

### Amazon DynamoDB

[Amazon DynamoDB](#) is a key-value database service. It delivers single-digit millisecond performance at any scale.

### Serverless

- DynamoDB is serverless, which means that you do not have to provision, patch, or manage servers.
- You also do not have to install, maintain, or operate software.

### Automatic Scaling

- As the size of your database shrinks or grows, DynamoDB automatically scales to adjust for changes in capacity while maintaining consistent performance.
- This makes it a suitable choice for use cases that require high performance while scaling.



**AMAZON RDS vs AMAZON DYNAMO DB**

# AmazonRDS

Automatic high availability;  
recovery provided



Customer ownership of data



Customer ownership of  
schema



Customer control of network



# Amazon DynamoDB

Key-value



Massive throughput  
capabilities



PB size potential



Granular API access



# Amazon Redshift

[Amazon Redshift](#) is a data warehousing service that you can use for big data analytics. It offers the ability to collect data from many sources and helps you to understand relationships and trends across your data.

## AWS DMS (Data Migration Service)

### Homogeneous migrations

can be from MySQL to Amazon RDS for MySQL, Microsoft SQL Server to Amazon RDS for SQL Server, or even Oracle to Amazon RDS for Oracle. The process is fairly straight forward.

Since schema structures, data types, and database code is compatible between source and target.

### heterogeneous migration

The second type of migration occurs when source and target databases are of different types.

This is called heterogeneous migration and it's a 2-step process.

Since the schema structures, data types, and database code are different between source and target, we first need to convert them using the AWS schema conversion tool.

This will convert the source schema and code to match that of the target database. The next step is then to use DMS to migrate data from the source database to the target database.

## Additional Database Services

## Amazon DocumentDB

[Amazon DocumentDB](#) is a document database service that supports MongoDB workloads. (MongoDB is a document database program.)

## Amazon Neptune

[Amazon Neptune](#) is a graph database service.

You can use Amazon Neptune to build and run applications that work with highly connected datasets, such as recommendation engines, fraud detection, and knowledge graphs.

## Amazon Quantum Ledger Database (Amazon QLDB)

[Amazon Quantum Ledger Database \(Amazon QLDB\)](#) is a ledger database service.

You can use Amazon QLDB to review a complete history of all the changes that have been made to your application data.

## Amazon Managed Blockchain

[Amazon Managed Blockchain](#) is a service that you can use to create and manage blockchain networks with open-source frameworks.

Blockchain is a distributed ledger system that lets multiple parties run transactions and share data without a central authority.

## Amazon ElastiCache

[Amazon ElastiCache](#) is a service that adds caching layers on top of your databases to help improve the read times of common requests.

It supports two types of data stores: Redis and Memcached.

## Amazon DynamoDB Accelerator

[Amazon DynamoDB Accelerator \(DAX\)](#) is an in-memory cache for DynamoDB.

It helps improve response times from single-digit milliseconds to microseconds.

