**Week-5 Day-1 Notes:**

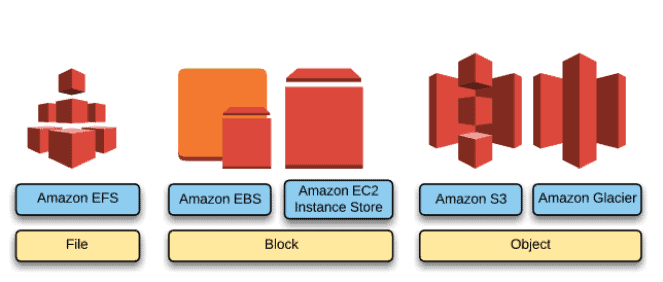
(Cloud Storage Service - AWS Storage Services - Amazon S3 - Section Introduction - S3 Buckets and Objects - Lab - S3 Buckets and Objects - S3 Versioning - Lab - S3 Versioning - S3 Encryption - Lab- S3 Encryption)

**Cloud storage** is a [cloud computing model](https://k21academy.com/amazon-web-services/aws-solutions-architect/cloud-service-models/) that stores data on the Internet through a cloud computing provider that manages and operates data storage as a service. It’s delivered on-demand with just-in-time capacity and costs and eliminates buying and managing your own data storage infrastructure.

**AWS Storage Services:**

AWS storage services are low-cost data storage with high durability and high availability. You also get the option to backing up information, archiving, and disaster recovery.

## **Storage Offered By Amazon Web Services (AWS)**

****

**1.  Simple Storage Service (Amazon S3)**

Amazon S3 the oldest and most supported storage platform of AWS uses an object storage model that is built to store and retrieve any amount of data. Data can be accessed everywhere such as websites, mobile apps, corporate applications, and data from IoT sensors or devices that can be dumped onto S3.

**Usage**

S3 has been highly used for hosting web content with support for high bandwidth and demand. Scripts can also be stored in S3 making it possible to store static websites that use JavaScript. It supports the migration of data to Amazon Glacial for cold storage, by using lifecycle management rule for data stored in S3.

## **2. Amazon Glacier**

Amazon Glacier is a secure, durable, and extremely low-cost storage service for data archiving and long-term storage. Glacier allows you to run powerful analytics directly on archived data. The glacier can also utilize other AWS storage services such as S3, CloudFront, etc. to move data in and out seamlessly for better and effective results.

### ****Usage****

Amazon Glacier is used to stores data in the form of archives. An archive can consist of a single file, or a combination of several files as a single archive, and the archives are organized in vaults. With support to querying on data to retrieve a particular subset of data that you need from within an archive.

## **3.  Elastic File System (Amazon EFS)**

As the name suggests EFS delivers a scalable, elastic, highly available, and highly durable network file system as-a-service. The storage capacity of EFS is elastic and is capable of growing and shrinking automatically depending on the requirement.

### ****Usage****

EFS is a network file system that can expand to petabytes with parallel access from EC2 instances. Elastic File System EFS is mounted on Amazon EC2 instances. Multiple EC2 instances can even share a Single EFS file system, allowing access to the large application that grew beyond a single instance.

## **4.  Elastic Block Store (Amazon EBS)**

Similar to EFS, EBS volumes are network file systems. Volumes get automatically replicated within Availability Zones for high availability and durability.

### ****Usage****

It is durable block-level storage to be used with EC2 instances in the [AWS cloud](https://k21academy.com/aws-training-courses/). EBS Volumes are used by mounting them onto EC2 instance as you will do with a physical hard drive on-premise and then format the EBS volume to the desired file system. EBS allows for dynamically increasing capacity, performance tuning and you can even change the type of volume with any downtime or performance impact.

## **5.  EC2 Instance Storage**

EC2 Instance storage provides temporary block-level storage for EC2 instances.

### ****Usage****

Instance storage volumes are ideal for the temporary storage of data that changes frequently like buffers, queue caches, and scratch data. It can only be employed by one EC2 instance meaning volumes can’t be detached and attached to a different instance.

## **6. Amazon FSx**

Amazon FSx is a completely managed third-party file system solution. Amazon FSx utilizes SSD storage to provide fast performance with low latency.

### Usage

With the use of Amazon FSx, you can utilize the rich feature sets and fast performance of widely-used open source and commercially licensed file systems, while avoiding time-consuming administrative tasks like hardware provisioning, software configuration, patching, and backups.

## **Benefits Of AWS Storage**

* No upfront cost it is a pay-as-you-go model.
* Worldwide access: You can access all your data worldwide just using an internet connection
* Storage can be increased or decreased with changes in data size.
* Low-cost data storage with high durability and high availability
* Plenty of choices for backing/archiving data in case of disaster recovery.

**Amazon S3 - Section Introduction**

Amazon S3 is an object storage service that stores data as objects within buckets. An *object* is a file and any metadata that describes the file. A *bucket* is a container for objects.

To store your data in Amazon S3, you first create a bucket and specify a bucket name and AWS Region. Then, you upload your data to that bucket as objects in Amazon S3. Each object has a *key* (or *key name*), which is the unique identifier for the object within the bucket.

S3 provides features that you can configure to support your specific use case. For example, you can use S3 Versioning to keep multiple versions of an object in the same bucket, which allows you to restore objects that are accidentally deleted or overwritten.

**S3 Buckets and Objects**

### Buckets

A bucket is a container for objects stored in Amazon S3. You can store any number of objects in a bucket and can have up to 100 buckets in your account. Every object is contained in a bucket.

For example, if the object named photos/puppy.jpg is stored in the DOC-EXAMPLE-BUCKET bucket in the US West (Oregon) Region, then it is addressable using the URL https://DOC-EXAMPLE-BUCKET.s3.us-west-.amazonaws.com/photos/puppy.jpg.

When you create a bucket, you enter a bucket name and choose the AWS Region where the bucket will reside. After you create a bucket, you cannot change the name of the bucket or its Region.

Buckets also:

* Organize the Amazon S3 namespace at the highest level.
* Identify the account responsible for storage and data transfer charges.
* Provide access control options, such as bucket policies, access control lists (ACLs), and S3 Access Points, that you can use to manage access to your Amazon S3 resources.
* Serve as the unit of aggregation for usage reporting.

**Objects**

Objects are the fundamental entities stored in Amazon S3. Objects consist of object data and metadata. The metadata is a set of name-value pairs that describe the object. These pairs include some default metadata, such as the date last modified, and standard HTTP metadata, such as Content-Type. You can also specify custom metadata at the time that the object is stored.

An object is uniquely identified within a bucket by a [key (name)](https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html#BasicsKeys) and a [version ID](https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html#BasicsVersionID) (if S3 Versioning is enabled on the bucket).

# **Steps to Create your first S3 bucket:**

1. Sign in to the AWS Management Console and open the Amazon S3 console at <https://console.aws.amazon.com/s3/>.
2. Choose **Create bucket**.

The **Create bucket** wizard opens.

1. In **Bucket name**, enter a DNS-compliant name for your bucket.

The bucket name must:

* + Be unique across all of Amazon S3.
  + Be between 3 and 63 characters long.
  + Not contain uppercase characters.
  + Start with a lowercase letter or number.

After you create the bucket, you cannot change its name.

1. In **Region**, choose the AWS Region where you want the bucket to reside.

Choose a Region close to you to minimize latency and costs and address regulatory requirements. Objects stored in a Region never leave that Region unless you explicitly transfer them to another Region.

1. Under **Object Ownership**, to disable or enable ACLs and control ownership of objects uploaded in your bucket, choose one of the following settings:

**ACLs disabled**

* + **Bucket owner enforced** – ACLs are disabled, and the bucket owner automatically owns and has full control over every object in the bucket. ACLs no longer affect permissions to data in the S3 bucket. The bucket uses policies to define access control.

**ACLs enabled**

* + **Bucket owner preferred** – The bucket owner owns and has full control over new objects that other accounts write to the bucket with the bucket-owner-full-control canned ACL.
  + **Object writer** – The AWS account that uploads an object owns the object, has full control over it, and can grant other users access to it through ACLs.

1. In **Bucket settings for Block Public Access**, choose the Block Public Access settings that you want to apply to the bucket.
2. (Optional) Under **Bucket Versioning**, you can choose if you wish to keep variants of objects in your bucket. To disable or enable versioning on your bucket, choose either **Disable** or **Enable**.
3. (Optional) Under **Tags**, you can choose to add tags to your bucket. Tags are key-value pairs used to categorize storage.

To add a bucket tag, enter a **Key** and optionally a **Value** and choose **Add Tag**.

1. (Optional) Under **Default encryption**, you can choose to configure your bucket to use server-side encryption . To disable or enable encryption, choose either **Disable** or **Enable**.
2. (Optional) If you want to enable S3 Object Lock, do the following:

a. Choose **Advanced settings**.

b. If you want to enable Object Lock, choose **Enable**, read the warning that appears, and acknowledge it.

1. Choose **Create bucket**.

**To upload an object to a bucket**

1. Open the Amazon S3 console at <https://console.aws.amazon.com/s3/>.
2. In the **Buckets** list, choose the name of the bucket that you want to upload your object to.
3. On the **Objects** tab for your bucket, choose **Upload**.
4. Under **Files and folders**, choose **Add files**.
5. Choose a file to upload, and then choose **Open.**
6. Choose **Upload**.

**S3 Versioning**

Versioning in Amazon S3 is a means of keeping multiple variants of an object in the same bucket. You can use the S3 Versioning feature to preserve, retrieve, and restore every version of every object stored in your buckets. With versioning you can recover more easily from user actions and application failures. After versioning is enabled for a bucket, if Amazon S3 receives multiple write requests for the same object simultaneously, it stores all of those objects.

Versioning-enabled buckets can help you recover objects from accidental deletion or overwrite. For example, if you delete an object, Amazon S3 inserts a delete marker instead of removing the object permanently. The delete marker becomes the current object version. If you overwrite an object, it results in a new object version in the bucket. You can always restore the previous version.

You can use S3 Versioning to keep multiple versions of an object in one bucket so that you can restore objects that are accidentally deleted or overwritten. For example, if you apply S3 Versioning to a bucket, the following changes occur:

* If you delete an object, instead of removing the object permanently, Amazon S3 inserts a delete marker, which becomes the current object version. You can then restore the previous version. For more information, see [Deleting object versions from a versioning-enabled bucket](https://docs.aws.amazon.com/AmazonS3/latest/userguide/DeletingObjectVersions.html).
* If you overwrite an object, Amazon S3 adds a new object version in the bucket. The previous version remains in the bucket and becomes a noncurrent version. You can restore the previous version.

**To enable or disable versioning on an S3 bucket**

1. Sign in to the AWS Management Console and open the Amazon S3 console at <https://console.aws.amazon.com/s3/>.
2. In the **Buckets** list, choose the name of the bucket that you want to enable versioning for.
3. Choose **Properties**.
4. Under **Bucket Versioning**, choose **Edit**.
5. Choose **Suspend** or **Enable**, and then choose **Save changes**.

**S3 Encryption**

Data encryption is the process of converting raw data into a coded form to help ensure that only authorized parties can read it. Encryption often uses a “key” (usually a large number) stored separately from the data to ensure that only the key holder can read it.

Data encryption protects your stored data against theft, ransomware attacks, and other security risks. If an attacker gets access or hold of your data, then they won’t be able to do anything with it unless they also get a hold of the key to unencrypt it.

**S3 Server-Side Data Encryption**

Within Amazon S3, Server Side Encryption (SSE) is the simplest data encryption option available. SSE encryption manages the heavy lifting of encryption on the AWS side, and falls into two types: SSE-S3 and SSE-C.

The SSE-S3 option lets AWS manage the key for you, which requires that you trust them with that information. With SSE-S3, you don’t have access to see or encrypt data using the key directly, but you can be assured that the raw data you own is encrypted at rest by AWS’s standard processes.

The SSE-C option similarly manages encryption and decryption of your data for you, but uses a key provided by you (the customer) and passed in to AWS with each request to encrypt or decrypt. AWS does not store your key with this method, so you are responsible for its safe keeping.

### ****S3 Client-Side Data Encryption****

[S3 Client-Side Encryption](https://docs.aws.amazon.com/AmazonS3/latest/dev/UsingClientSideEncryption.html) puts all the responsibility for the encryption heavy lifting onto the user. Rather than allowing AWS to encrypt your data, you perform the encryption within your own data center and upload the encrypted data directly to AWS.

Small numbers of objects or single files may be encrypted one at a time in the Amazon S3 console.

1. Sign into the AWS Management Console.
2. Navigate to the S3 console and find the bucket and object that was flagged as unencrypted.
3. Select the object and choose Properties then Encryption.
4. Use the wizard to choose the S3 encryption options you prefer.
5. Save to apply encryption to the object.

S3 Security & Bucket Policies - Lab - S3 Security & Bucket Policies