

# Task #4

## ex#1

### Object storage

Object storage stores data as independent objects that are accessed through an API, without a real folder structure. It is suitable for cloud environments and for storing large amounts of unstructured data.

#### Advantages:

- Can easily handle growth in data and users (**scalability**)
- High reliability due to data replication
- Cost-effective for large files and backups

#### Disadvantages:

- Does not support partial file updates
- Slower access compared to block storage

### File storage

File storage uses a traditional file and folder structure that is shared between multiple machines. Applications can access it like a regular filesystem.

#### Advantages:

- Familiar structure and easy to use
- Supports partial file reads and writes

#### Disadvantages:

- Harder to handle large growth (**scalability**)
- Less suitable for cloud-native systems

## Block storage

Block storage presents data as virtual disks attached to servers and is mainly used for performance-critical systems.

### **Advantages:**

- Fast access and high performance
- Suitable for databases

### **Disadvantages:**

- Limited ability to grow easily (**scalability**)
- Usually attached to a single server

## ex#2

S3 (Amazon Simple Storage Service) is a cloud storage service provided by Amazon Web Services (AWS). The service allows you to store and retrieve data widely, especially for large files like images, videos, backups, logs, and more.

### Benefits of S3

- **Scalability** – You can expand storage almost without limit, according to your needs.
- **High Availability** – The service allows access to data at any time and from anywhere.
- **Security** – You can set access permissions and security measures like encryption.
- **Widespread Use** – Many companies use S3 to store information, backups, and transfer files over the internet.

In general, if you need to store a large amount of data easily, with flexibility and the ability to scale as needed, S3 is an excellent service.

## ex#3

A **Bucket** in Amazon S3 is a container for storing files (objects). It's like a folder, but with a few differences:

1. **Unique Name** – Every Bucket must have a unique name worldwide.
2. **Data Storage** – You store files (images, videos, documents, etc.) inside it.
3. **Folder Hierarchy** – You can organize files inside the Bucket like folders by using "/" in file names.
4. **Access Permissions** – Each Bucket has its own access permissions that you can configure.

In short, a **Bucket** is a place where you store your files in S3, with unique names and independent access settings.

## ex#4

No, in S3 there are no **real folders** like in traditional file systems. However, you can **simulate folders** by using `/` in the file names. This allows for easy and organized structuring of files, and it's also efficient for later retrieval based on categories (like `folder1/`). This way, you can easily organize files and retrieve them in an orderly fashion based on their names.

## ex#5

Yes, S3 has limitations, but they are different from traditional file systems:

- **Individual File Size:** Up to 5 terabytes per file, with no limit on the total size of the bucket or the number of files.
- **Limitations in traditional file systems:** There are file size limitations (usually up to a few terabytes), and physical disk size limitations.
- **Scalability:** S3 offers scalable storage, while traditional file systems have physical limits.
- **Performance:** S3 may have potential latencies due to communication with AWS servers, while local file systems offer direct and faster access.

## ex#6

What implementations of S3 exist?

There are several **S3-like services** and **object storage solutions** that are compatible with the S3 API. Here are some examples:

1. **Amazon S3** – The official object storage service from Amazon.
2. **MinIO** – An open-source object storage solution compatible with S3, allowing you to set up a local or private storage server.
3. **Wasabi** – A cloud storage service offering a cheaper alternative to S3 with S3 API support.
4. **DigitalOcean Spaces** – An object storage service from DigitalOcean that supports the S3 API.
5. **Backblaze B2** – An object storage service compatible with S3, offering a low-cost cloud storage solution.
6. **IBM Cloud Object Storage** – IBM's object storage service that supports the S3 API.
7. **Google Cloud Storage** – Google's cloud storage service that supports the S3 API and provides similar object storage solutions.

These services support the S3 API, so you can use them in a similar way to Amazon S3, with additional features or benefits such as lower costs or tailored performance.

## ex#7

```
C:\Users\876\Downloads\task4\ex7\minio\data>docker run -d -p 9000:9000 -p 9001:9001 --name minio -v C:\Users\876\Downloads\task4\ex7\minio\data:/data -e MINIO_ROOT_USER=NoaK -e MINIO_ROOT_PASSWORD=@4110803 minio/minio server /data --console-address ":9001"
b2c738bc62199725bf68b9ab27ec3f1806a2354cf05b6e8e7c2351807588a97f

C:\Users\876\Downloads\task4\ex7\minio\data>docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
b2c738bc6219        minio/minio      "/usr/bin/docker-ent..."   9 seconds ago     Up 8 seconds      0.0.0.0:9000-9001->9000-9001/tcp, [::]:9000-9001->9000-9001/tcp    minio
```

The screenshot shows the MinIO Object Store login page. The header reads "MINIO OBJECT STORE Community Edition". Below the header, there is a "High-Performance Object Store" banner with a dark background and blue text. The banner contains the following text:  
MinIO is a cloud-native object store built to run on any infrastructure: public, private or edge clouds, bare metal, racks, and hybrid environments. It is a fast, reliable, SaaS applications and fast backup & recovery. MinIO is dual-licensed under GPLv3 and AGPL v3 and commercial license. To learn more, visit [www.min.io](http://www.min.io).

The login form has two fields: "NoaK" in the username field and "\*\*\*\*\*" in the password field. There is also a "Remember Me" checkbox. At the bottom of the form is a "Login" button.

At the very bottom of the page, there is a footer with links: "Documentation | GitHub | Support | Download".

The screenshot shows the MinIO Object Browser interface. At the top left is the 'MINIO OBJECT STORE Community Edition' logo. The main title 'Object Browser' is centered above a search bar containing the placeholder 'Start typing to filter objects in the bucket'. To the right of the search bar are two small icons: a question mark and a star. On the far left, a sidebar contains links for 'Create Bucket', 'Filter Buckets' (which is currently selected), 'Buckets', and 'task4'. The 'task4' item is highlighted with a blue background. The main content area shows the 'task4' bucket details: 'Created on: Thu, Jan 01 2026 08:22:44 (GMT+2)' and 'Access: PRIVATE'. Below this are three buttons: 'Rewind' with a circular arrow icon, 'Refresh' with a circular arrow icon, and 'Upload' with an upward arrow icon. A navigation bar at the bottom left includes a back arrow, the path 'task4', and a folder icon. To the right of the path is a 'Create new path' button with a folder icon. The central message in the content area reads 'This location is empty, please try uploading a new file'.

MINIO  
OBJECT STORE  
Community Edition

Create Bucket

Filter Buckets

Buckets

task4

Documentation

License

Sign Out

Object Browser

Start typing to filter objects in the bucket

task4

Created on: Thu, Jan 01 2026 08:22:44 (GMT+2) Access: PRIVATE

< task4 / test.txt.txt Create new

Name	Last Modified	Size
test.txt.txt	Today, 08:25	-

Downloads / Uploads

test.txt.txt

Bucket: task4

100%

Delete

## ex#8

### The code has been uploaded to Git

The run:

```
● PS C:\Users\876\Downloads\task4\ex8> python minio_client.py
Created object: file_567.txt with content: wyHk860WsH3owObc6wI8
Created object: file_523.txt with content: dAmg2XHS6daQFo6Ynb2q
Objects in bucket:
- file_523.txt
- file_567.txt
- file_gkkzh.txt
- file_z9ip0.txt
- test.txt.txt
reading data of object: file_567.txt
Data: wyHk860WsH3owObc6wI8
Deleted object: file_523.txt
reading data of updated object: file_567.txt
Data: Updated content1: uyz8DuqsXwdaicfSSKS8
○ PS C:\Users\876\Downloads\task4\ex8>
```

After the run:

The screenshot shows the MinIO Object Browser interface. On the left, there's a sidebar with various icons. The main area displays the 'Object Browser' for a bucket named 'task4'. The bucket was created on Thursday, January 01, 2026, at 08:22:44 (GMT+2). It has a PRIVATE access level and contains 4 objects, totaling 114.0 B.

The object list table has columns for Name, Last Modified, and Size. The objects listed are:

Name	Last Modified	Size
file_567.txt	Today, 09:34	38.0 B
file_gkkzh.txt	Today, 09:31	38.0 B
file_z9ip0.txt	Today, 09:34	38.0 B
test.txt.txt	Today, 08:25	-

To the right of the table, there's a sidebar titled 'Actions:' which includes options like Download, Share, Preview, Tags, and a 'Delete' button. Below the actions is a 'Display Object Versions' link. At the bottom right, there's an 'Object Info' section with a magnifying glass icon.