# Pros

## Scaling

It is easier to scale (add more nodes) compared to for example HDFS.

## Fault tolerance

Data is replicated what ensures availability and durability.

## Easy of setup

It is easy to set up.

## Decoupled storage and compute

Compute programs (like Spark) can access object storage data over HTTP(S) — even across different clusters or clouds.

You can easily scale:

* Storage: by adding nodes with more or cheaper disks
* Compute: by adding CPU/GPU-heavy nodes or autoscaling pods

This works well because storage and compute are fully decoupled.

## Accessing data from outside of a cluster

Accessing data from outside of a cluster where we run our object storage system, for example for reporting, is easy through HTTPS (that’s related to decoupling storage and compute).

## Efficient for storing big number of small files

When we have a lot of small files then object storage is very efficient for storing them. It is much better than HDFS.

# Cons

## Speed

Reading / writing data is slower compared to for example HDFS. That’s because data is sent through a network using HTTPS while in HDFS it is accessible locally through TCP.