# Introduction

This is a tool for object storage. It can be well managed on Kubernetes.

# Comparison to HDFS

## Separation of storage and compute

With miniIO it is easier to scale separately storage and compute. That’s because when you have data on one server and compute engine (like Spark) on another, then with miniIO it is not a problem.

Compute engine can get data stored on another server with miniIO efficiently over a network. On the other hand getting data from another server stored with HDFS is much less efficient because of networking design used for that.

## Speed of reading / writing data

When we have data stored in HDFS cluster, and we run for example Spark on the same cluster, then reading and writing data is faster than with miniIO.

But when Spark needs to read data from a server different then the one running Spark, then reading data from miniIO is faster then from HDFS.

It might be a good idea to keep all the historical data in the miniIO and in the HDFS cluster keep only data from the recent few years, which is currently used in analytics, and run Spark on the same cluster as HDFS for fast reads/writes.