

Введение

Возьмем для примера такую задачу: загрузить и сохранить аватарку пользователя. Для этого нужно:

1. отправить картинку на сервер
2. как-то обработать ее
3. сохранить на диск
4. настроить веб-сервер, который сможет отдать клиенту эту картинку по ссылке

На этом список подзадач заканчивается, но возникают вопросы. Что делать, если:

- нагрузка выросла, и для обработки запросов нужно несколько серверов?
- картинок очень много и памяти сервера не хватает?
- пользователей очень много, и мы не можем с нужной скоростью отдавать им картинки?
- пользователи географически распределены по миру? Например, наши сервера находятся в России, а клиенты в Америке. Из-за этого возникает большая задержка при передаче данных.
- приложение запускается в оркестраторе (например, kubernetes.io) и требует не хранить данные, то есть быть [stateless](https://en.wikipedia.org/wiki/Stateless).

Решением будет централизованное хранилище, которые поддерживает масштабирование, резервирование и будет географически распределено по миру. Позволить себе такое могут только большие корпорации. Это они и сделали, реализовав объектные хранилища, которые решают все проблемы выше:

- <https://aws.amazon.com/ru/s3/>
- <https://www.digitalocean.com/products/spaces/>
- <https://mcs.mail.ru/storage/>
- <https://sbercloud.ru/ru/products/object-storage>
- <https://selectel.ru/services/cloud/storage/>
- <https://cloud.yandex.ru/services/storage>
- и т.д.

У каждого облачного провайдера есть свое хранилище, но работают они по общему протоколу — [Amazon S3](https://aws.amazon.com/s3/).