

Предположим, мы стоим перед задачей создания системы для хранения и обмена большим объемом неструктурированных данных, таких как изображения, видео, аудиофайлы, документы и т.д. Наши ключевые требования к системе управления данными могут быть следующими:

- Масштабируемость: Нам необходима система, которая легко и эффективно масштабируется под возрастающий объем данных.
- Быстрый доступ: Пользователям требуется быстрый и надежный доступ к хранящимся файлам в любое время и с любого места.
- Безопасность: Данные должны быть надежно защищены от несанкционированного доступа и потерь.
- Работа с большим объемом файлов: Нам требуется система, которая может эффективно работать с большим количеством файлов.

В этом случае, облачное **NoSQL типа файловое хранилище, такое как Amazon S3 или MinIO**, может быть идеальным выбором. Эти системы обеспечивают высокую масштабируемость, быстрый доступ и высокую безопасность данных. Они также хорошо справляются с большим количеством файлов и позволяют управлять доступом к данным на уровне отдельных файлов или бакетов.

С другой стороны, цена использования облачных хранилищ может быть довольно высока, особенно при работе с большим объемом данных и активной загрузкой и скачиванием файлов. Кроме того, может потребоваться дополнительное усилие для интеграции с существующими системами и поддержания безопасности данных. В этих случаях, локальное хранилище или более специализированное решение может быть лучшим выбором.

В файловой базе данных типа Amazon S3, данные обычно представлены в виде объектов внутри "корзин" (buckets). Каждый объект имеет уникальное имя (ключ) и содержимое (значение, которое может быть файлом).

Bucket: my-bucket

```
|
|— file1.txt
|
|— directory1
|   |— file2.jpg
|   └─ file3.png
|
└─ directory2
    |— file4.mp3
    └─ directory3
        └─ file5.docx
```

Есть и **другие NoSQL решения** (графовые, например) но они используются реже.