

Архитектура вычислительных систем

Лекция 7. Слой хранения
Часть 3



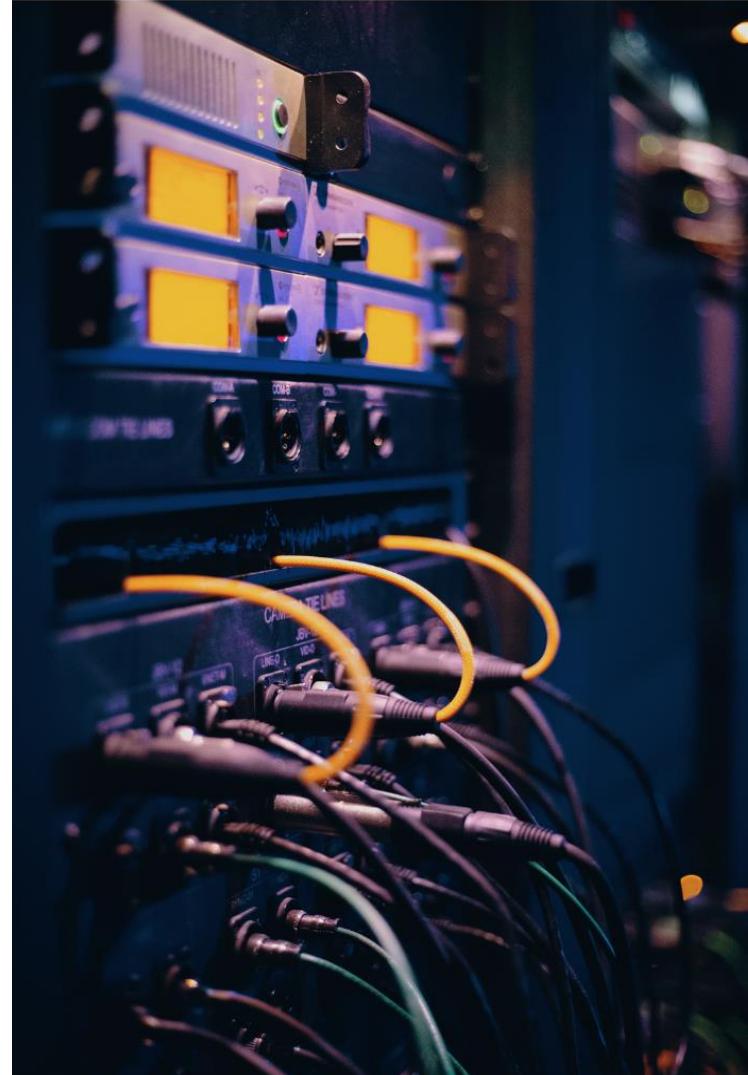
Artem Beresnev

t.me/ITSM Dao

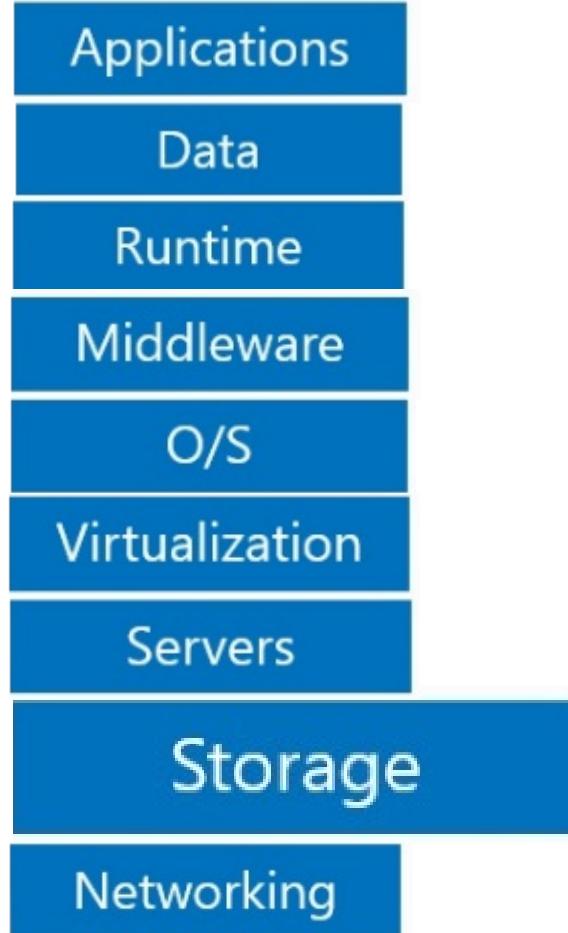
t.me/ITSM Dao Chat

Слои ИТ-инфраструктуры

Что там было про хранение?

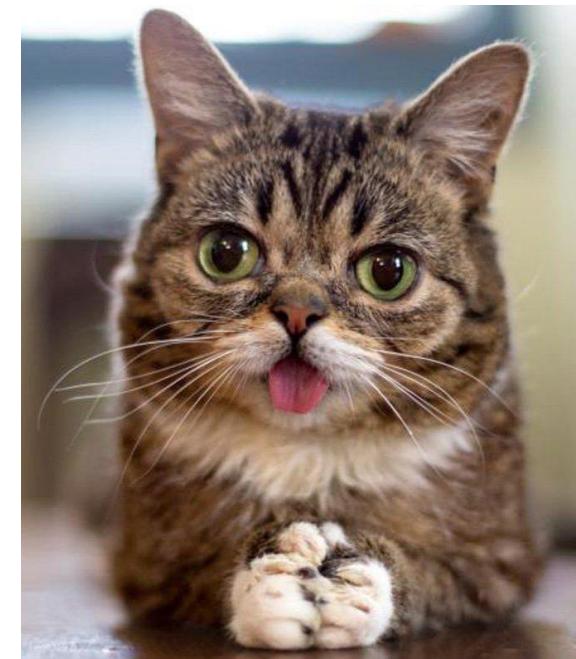


Слой Storage. Задачи



- Хранение данных
- Обеспечение:
 - **надежности**
 - **производительности с учетом требований приложений**
 - **версионирования**
 - **масштабирование**
 - безопасности
 - экономической эффективности

ОБЪЕКТНОЕ ХРАНЕНИЕ

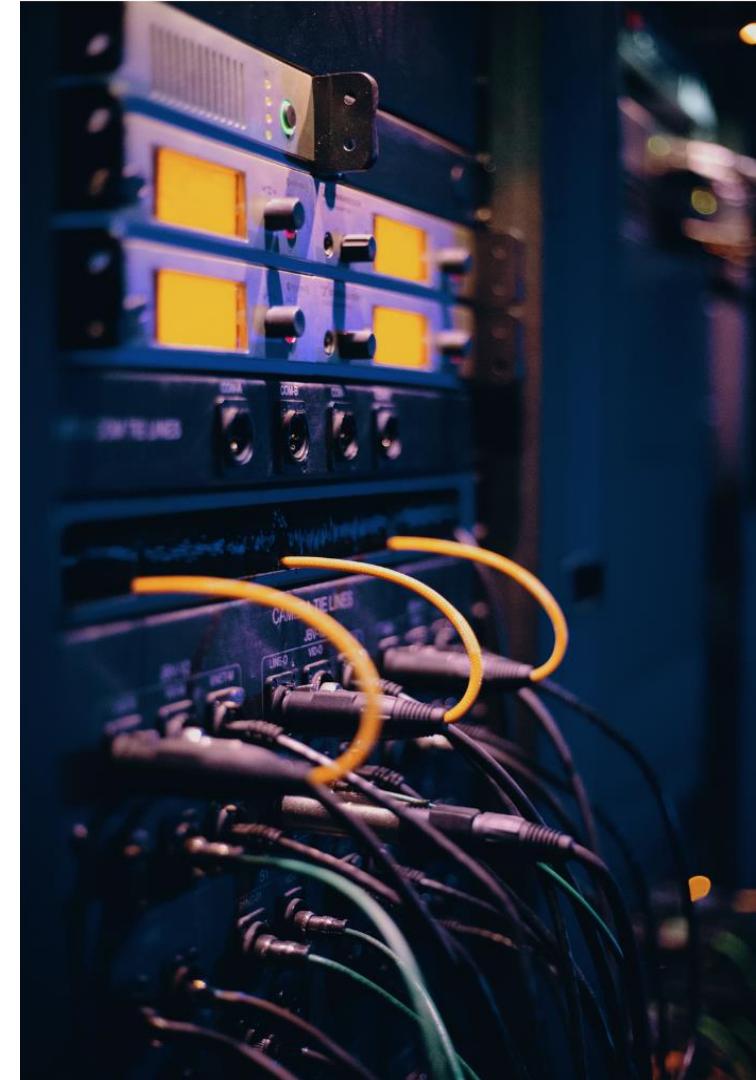


Классификация

- **Блочное хранение**
диски, DAS, SAN
- **Файловое хранение**
файловые системы, NAS
- **Объектное хранение**
S3

S3

Самое ценное в современных ИС – данные. На слое Storage существует множество техник обеспечения избыточности

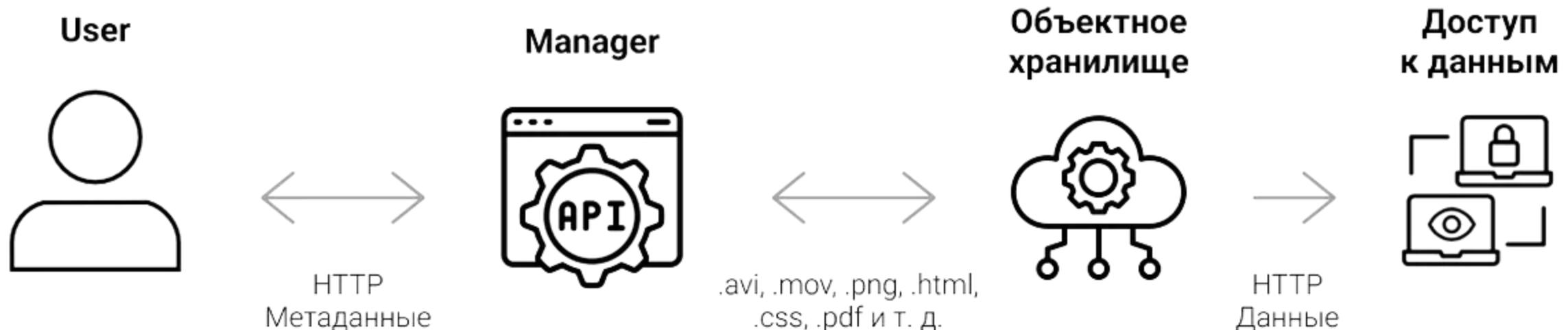


S3 (Simple Storage Service)

S3 (Simple Storage Service) — это объектное хранилище для неструктурированных данных любого типа и объема, таких как изображения, видео, логи и архивы.

В отличие от файловых систем с иерархической структурой папок, S3 использует плоскую модель, где каждый объект идентифицируется уникальным ключом без жесткой вложенности.

Это обеспечивает высокую масштабируемость и простоту доступа через REST API.



S3 – это архитектура и API

Open Source S3-совместимые хранилища

MinIO: высокопроизводительное объектное хранилище для on-premise и контейнеров, полностью совместимо с S3 API

Ceph (RADOS Gateway): распределенная система с S3-совместимостью через RGW, подходит для крупных развертываний, масштабируется от нескольких до тысяч узлов.

Scality S3server: легковесное решение для тестирования и разработки S3-приложений локально или в Docker.

s3fs: FUSE-файловая система для монтирования S3-бакетов как локального диска в Linux/macOS.

Облачные S3-совместимые хранилища

Amazon S3: оригинальное хранилище AWS с полным API, классами хранения и глобальной доступностью.

Yandex Cloud Object Storage: S3-совместимое, интегрируется с российскими сервисами, поддержка lifecycle и шифрования.

VK Cloud S3: объектное хранилище с S3 API для бэкапов, медиа и данных ML.

Selectel Object Storage: Tier III data-центры, соответствует 152-ФЗ, горячее/холодное хранение.

Cloud4Y S3: российский провайдер с прозрачной тарификацией и высокой скоростью.

Cloudflare R2: S3-совместимое без платы за egress, фокус на низкие затраты и миграцию

Что получаем?

- Доступ по HTTP/HTTPS
- Аутентификация через ключи доступа
- Операции PUT (загрузка), GET (скачивание), DELETE и LIST.
- Автоматическое горизонтальное масштабирование
- Надежность (данные реплицируются на нескольких узлах, защита от сбоев).
- Интеграция с облачными сервисами (EC2, Lambda),
- Версионирование
- Шифрование
- lifecycle-политики для автоматизации удаления.
- Оплата по факту использования



Сущности S3

Бакет (Bucket) — логический контейнер для хранения объектов с уникальным глобальным именем. Нет иерархии.

Объект (Object) — базовая единица данных, включающая файл (до 5 ТБ), метаданные (ключ-значение) и уникальный ключ (путь, например, prefix/file.txt). Объекты атомарны, реплицируются автоматически для высокой доступности.

Ключ (Key) — уникальный идентификатор объекта в бакете, строка UTF-8 до 1024 байт, определяющая "путь" без реальной вложенности.

Префикс (Prefix) — начальная часть ключа для группировки объектов, имитирует папки (например, logs/2025/).

Классы хранения — варианты для оптимизации затрат. Горячий, теплый, холодный



Чем работать с S3?

Консоль управления: веб-интерфейс (AWS Console, Yandex Cloud) для GUI-операций.

s3cmd/rclone: CLI-утилиты для Linux, поддержка скриптов и мониторинга как FS (rclone sync local/ remote:bucket).

CLI: командная строка для Linux/Windows (aws s3 ls s3://bucket, aws s3 cp file s3://bucket/key), требует конфигурации ключей

SDK: библиотеки для языков (boto3 для Python, AWS SDK для Java/JS), интеграция в приложения.

Terraform/Ansible: IaC для автоматизации создания бакетов и политик, подходит для DevOps.



Чем работать с S3? Примеры

Список бакетов:

```
aws s3 ls --endpoint-url=https://storage.yandexcloud.net
```

Список объектов в бакете:

```
aws s3 ls s3://my-bucket/ --endpoint-url=https://storage.yandexcloud.net
```

Создание бакета:

```
aws s3 mb s3://my-new-bucket --endpoint-url=https://storage.yandexcloud.net
```

Рекурсивная загрузка:

```
aws s3 cp mydir/ s3://my-bucket/ --recursive --endpoint-url=https://storage.yandexcloud.net
```

Скачивание файла:

```
aws s3 cp s3://my-bucket/key.txt localfile.txt --endpoint-url=https://storage.yandexcloud.net
```

Синхронизация:

```
aws s3 sync localdir/ s3://my-bucket/ --endpoint-url=https://storage.yandexcloud.net
```