

ИНТЕРНЕТ-СЕРВИС ДЛЯ ЗАГРУЗКИ И СОВМЕСТНОГО ПРОСМОТРА МЕДИА

Мое приложение – это социальная сеть, которая позволяет пользователям создавать и делиться виртуальными досками, на которых могут храниться изображения, видео, ссылки и другие объекты, связанные с определенной темой. Пользователи могут искать и сохранять контент, создавать собственные коллекции и делиться ими с другими пользователями. Приложение – это место, где пользователи могут найти вдохновение и идеи для различных проектов и задач. Кроме того, такое приложение предоставляет бизнесу возможность продвигать свои продукты через создание своих профилей, а также может быть использовано для поиска информации и различных ресурсов.

Главными конкурентами моего приложения являются Pinterest и TikTok, поэтому лучше всего сравнить с этими двумя приложениями. Главная проблема Pinterest и TikTok – это отсутствие приватных групп, слабое разделение контента по категориям и отсутствие улучшения качества с помощью нейросетей. Мое приложение решает все эти проблемы, а также добавляет улучшенные рекомендации контента, встроенное улучшение качества и фильтры на основе нейросетей.

Для разработки проекта была выбрана облачная инфраструктура. В последнее время данная инфраструктура набрала огромную популярность среди разного масштаба бизнесов из-за возможности быстрой масштабируемости, высокого уровня безопасности и снижения затрат поддержку и обслуживание. Также для разработки была выбрана бессерверная архитектура (serverless-архитектура) [1]. Выбор лежал между бессерверной и микросерверной архитектурами. Бессерверная архитектура обычно лучше подходит для приложений, которые не требуют сложной логики или масштабирования на большие объемы. Она может облегчить развертывание и управление приложением, а также снизить затраты на инфраструктуру. Микросервисная архитектура обычно лучше подходит для более сложных приложений, которые требуют интенсивного использования ресурсов и быстрого масштабирования. Она может обеспечить более гибкое и масштабируемое приложение, а также упростить его сопровождение и развитие. Однако, микросервисная архитектура может быть более сложной в управлении и требует большей инфраструктуры, что может увеличить затраты на приложение. Поэтому для разработки данного приложения был выбран подход бессерверной архитектуры. Все паттерны разра-

ботки с использованием облаков прописаны в документации Well-Architecture Principles [2].

В качестве провайдера облака был выбран Google Cloud Platform (GCP) [3]. Далее описаны все сервисы, которые будут использоваться для разработки приложения, а также как они используются в данном проекте. Сервис Cloud Storage будет использоваться для хранения всех изображений и видео приложения, а также необработанных журналов и отчетов приложения. Сервис Cloud SQL будет использоваться для OLTP базы данных приложения. BigQuery – это хранилище исторических данных и журналов приложения для дальнейшего анализа и создания отчетов. Cloud Pub/Sub будет использоваться для нескольких целей: передача видео-контента для последующей обработки, передача параметров на облачные функции, а также передачи данных для длительной обработки. Через облачные функции будут реализованы все функции, к которым будет обращаться front-end приложения, а также некоторые сервисы для работы с БД, с облачным хранилищем и запуском ETL конвейерных функций.

Главная проблема в разработке приложения возникла при обработке видео и его трансформации с помощью нейросетей. Этот процесс затратный и продолжительный время. Процесс обработки видео следует разбить на несколько этапов, таких как разбивка видео на кадры, обработка каждого кадра, склейка кадров в единое видео и т. д. Каждый этап выполняется с помощью отдельных облачных функций, и сообщения о наступлении следующего этапа могут быть переданы с использованием сервиса издатель/подписчик Pub/Sub. Например, можно создать очередь сообщений для разбивки видео на кадры, и внутри облачной функции обрабатывать кадр, и когда задача будет выполнена, каждый кадр может быть отправлен в другую очередь сообщений для соединения в полноценное видео. Главное преимущество такого подхода в том, что этот процесс можно разбить на несколько облачных функций для распределения нагрузки, главное при передаче кадра в очередь сообщений указывать порядковый индекс для восстановления порядка кадров в видео.

ЛИТЕРАТУРА

1. Serverless Architecture [Электронный ресурс] – Режим доступа: <https://www.datadoghq.com/knowledge-center/serverless-architecture/>
2. AWS Well-Architected [Электронный ресурс] – Режим доступа: <https://aws.amazon.com/ru/architecture/well-architected/>
3. Google Cloud Docs [Электронный ресурс] – Режим доступа: <https://cloud.google.com/docs>