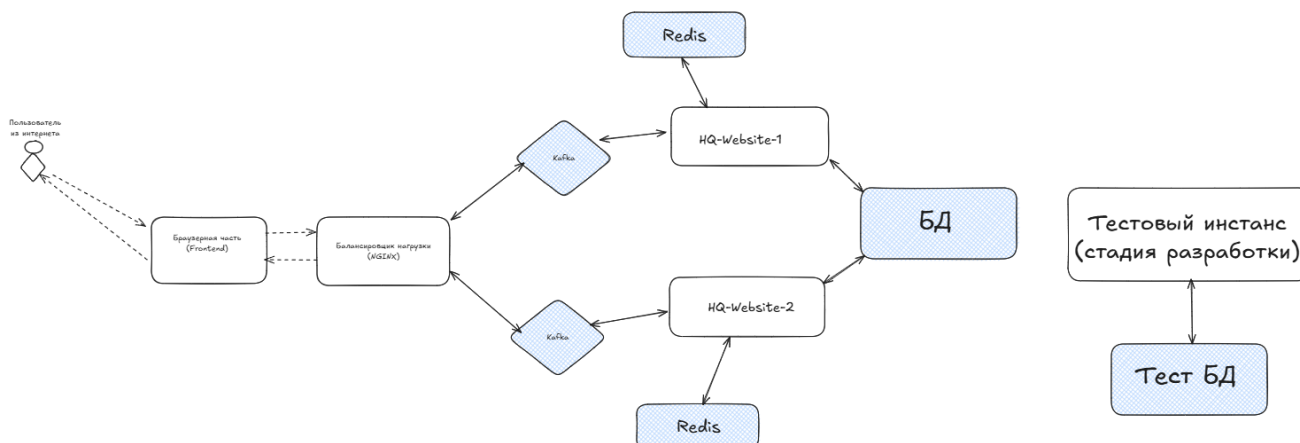


# Развитие, проекты

## Архитектура веб-приложения

Архитектурный подход к разработке: **Микросервис**



Такой подход позволяет разрабатывать приложение как можно быстрее, за счёт того, что не нужно думать о взаимодействии компонентов по сети, так как одно приложение - один сервер.

На схеме изображен примерный план по внедрению, где 1 геометрический объект - 1 сервер. Стрелками указано их взаимодействие друг с другом.

### Условные обозначения:

**Frontend** - то, что видит пользователь, перейдя по ссылке в интернете. Все взаимодействие пользователя с серверной частью происходит именно там.

**NGINX** - Балансировщик нагрузки, так же - обратный прокси. Служит для обработки запросов из интернета и распределению их по серверам.

**Kafka** - брокер сообщений, служит для обеспечения надежности при передаче критически важных данных.

**Redis** - база данных для хранения пользовательских сессий. Нужно для кэширования неизменяемых данных на веб-странице, которые пользователь уже получал из БД, для ускорения загрузки веб-страницы у пользователя и для снижения нагрузки на основной сервер БД.

# Развитие, проекты

**Hq-Website-<n>** - Сервер с веб-приложением для производственной среды. На этом сервере будет развернут наш монолит. Рекомендуется устанавливать как минимум 2 таких сервера, для того, чтобы обеспечить надежность и устойчивость к разного рода сбоям.

**БД (База данных)** - база данных на которой хранятся все основные данные веб-приложения. Для получения конкретики о том, какие именно хранятся данные в БД, обратитесь к схеме разработки сущностей и связей базы данных.

**Тестовый инстанс** - изолированный экземпляр веб-приложения для разработки нового функционала.

**Тест БД** - изолированный экземпляр базы данных для тестового инстанса веб приложения. Наполняется тестовыми данными, для того, чтобы не затрагивать основную БД при разработке нового функционала.

## Или вариант проще

Если мы хотим рассмотреть минимальный возможный вариант для запуска проекта, то можно использовать следующую схему:



При минимальных нагрузках и отсутствии передачи критических данных, которые нельзя потерять (на начальном этапе) это вполне подойдет

Уникальный ID ответа: #1368

Опубликовал: : Алексей Данилов

Последнее обновление: 2025-05-16 10:00