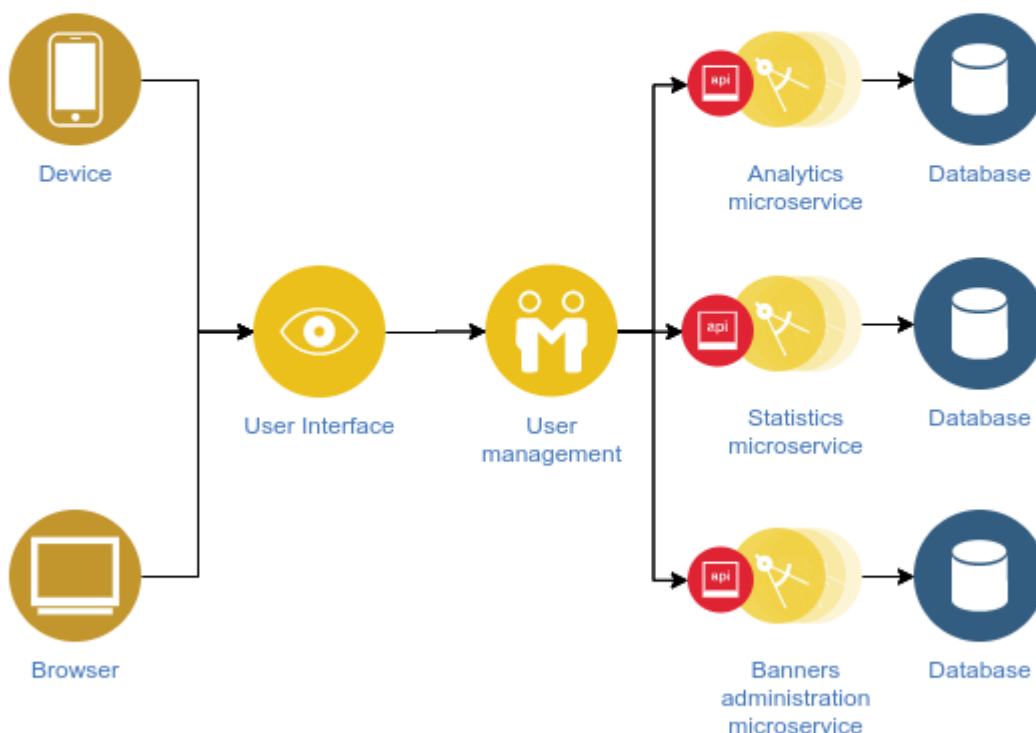


Саенко Богдан

Тестовое задание для поступления на курс Highload Software Architecture

– опишите высокоуровневую архитектуру проекта;



– выделите ключевые компоненты / микро сервисы;

1. User Interface
2. Управление пользователями и их данными
3. API для статистических данных
4. API для аналитических данных
5. API для управления рекламными баннерами

– оцените RPS если ваши баннера будут размещены на top 10 сайтах в UK (по одному баннеру на страницу);

$$RPS = \frac{1000}{T} \times workers, \text{ где } T - \text{среднее время обработки запроса (в миллисекундах), workers}$$

– количество потоков.

Согласно данным сайта littledata.io, по результатам исследования проведенного в мае 2020 года и включающее в себя 3560 сайтов с посещаемостью от 20 до 500 тысяч посещений в месяц, среднее время обработки запроса составляло 615 мс. Если время обработки запроса меньше 324 мс, то наш сайт входит в 20% лучших, а если меньше 220 мс, то в 10%. [Ссылка на исследование](#)

Допустим, что мы стремимся попасть в число лучших и достигнем показателя 200 мс для нашего сайта.

Пользуясь данными сайтов similarweb.com возьмем статистику по 10 самым посещаемым сайтам UK. Ссылка на статистику: <https://bit.ly/39AqOIS>

На сайте есть общие данные по посещениям за октябрь 2020 года и процентные данные по странам. После несложных вычислений можно узнать данные по просмотрам за секунду для каждого сайта для UK.

Данные вынесем в таблицу:

	Сайт	Total Visits per second (average)	Pages per Visit
1	google.com	1372	8.52
2	youtube.com	493	11.28
3	facebook.com	379	8.46
4	amazon.co.uk	155	8.16
5	twitter.com	146	11.97
6	google.co.uk	140	10.00
7	bbc.co.uk	228	3.25
8	ebay.co.uk	110	10.42
9	wikipedia.org	118	3.00
10	live.com	76	8.15
	Total	3217	83.21

Представим ситуацию в которой все пользователи, зашедшие на данные сайты переходят по нашему баннеру и рассчитаем *RPS* для данного случая:

$$RPS = \frac{1000}{200\text{ ms}} \times 3217 = 16085$$

Данный случай изначально является фантастическим, поэтому, считаю, что мы можем смело брать 15-20% от этого значения и поднимать инфраструктуру исходя из него.

– как вы организуете систему таргетинга?

Если имеется в виду настройка таргетинговой системы для продвижения нашего аналога AdSense, то параметры будут следующие:

1. Возраст: 16-45;
2. Пол: любой
3. География: по сути весь англоязычный мир, но если мы из UK, то можно основной акцент сделать на UK и страны Евросоюза, возможно еще США, Канада;
4. Ключевые слова: Internet advertising, contextual advertising UK, contextual advertising, advertising on the site.

Если имеется в виду, как мы будем настраивать наш продукт, чтобы показывать рекламу целевой аудитории, которая нужна нашим заказчикам, то мы будем запрашивать файлы cookies у посетителей, собирать с них данные, анализировать и показывать соответствующую рекламу.

– какие технологии предложите?

При сборке предлагаю каждый микросервис упаковывать в Docker image и сохранять их в Docker Registry. Сборку так же необходимо автоматизировать, используя один из следующих инструментов (Jenkins, Gitlab CI, Github Actions).

Для развертывания всей нашей микросервисной архитектуры предлагаю использовать Kubernetes для оркестрации. Так же, если позволяет бюджет, предлагаю развернуть Kubernetes кластер в cloud (AWS (EKS), Google (GKE)) для простоты развертывания, мейнтейнинга и пр. Так же для простоты и при необходимости автоматического

разворачивания Kubernetes кластера, предлагаю использовать Infrastructure as a Code инструмент – Terraform (для удобства работы в команде предлагаю использовать Terraform Cloud, если позволяет бюджет).

Upd. Данное предложение формировалось до анонса Kubernetes 1.20. Для контейнеризации необходимо рассмотреть так же другие технологии, а не только Docker, возможно другие решения подойдут нам больше.

– как будете трекать статистику?

В блоке кода, который будет вставляться на сайте, будет счетчик. Как он будет работать? Когда кто-то нажимает на баннер, отправляется запрос в базу данных, в специальную таблицу, в которой создается запись, что был совершен клик. Так же указывается время когда этот клик был произведен, геолокация, источник (на каком сайте “кликнули” на объявление).

Таким образом мы сможем увидеть сколько в сумме людей перешло по этому баннеру, а также, пользуясь разными фильтрами при запросах, мы сможем узнать с какого сайта сколько людей перешло, когда осуществлялись переходы и откуда. Для начального запуска проекта, по моему мнению, этой информации будет достаточно, дальше уже на этой основе можно добавлять еще другие данные для записи в таблицу.

Для каждого клиента нашей компании будет создаваться отдельная БД, для каждого баннера этого клиента – отдельная таблица, в которую будет записываться статистика по конкретному баннеру.

– где будете хранить статистику по показам/кликам?

Хранить статистику будем в базе данных. К сожалению, пока не знаю какая именно база данных будет лучшей для нашего случая, но в описании курса написано, что на курсе мы с этим вопросом разберемся)) Так же для баз настроим репликации и back-up процесс с возможностью хранения бекапов на удаленном хранилище (к примеру, AWS S3).

– оцените опенитировочный Total Cost of Ownership.

Все зависит от того на каком этапе развития находится наша компания, потому что при старте будут одни нагрузки, требования по инфраструктуре и масштабированию, а на следующих этапах – другие. Для начального развертывания нам необходимо 3 Kubernetes кластера для трех окружений (Development, Stage, Production), которые мы будем крутить в AWS:

$EKS - 0.1 \$ \text{ per hour} * 24 * 30 * 3 \text{ clusters} = 216 \$ \text{ per month}$

Для каждого кластера будем использовать по 3 Kubernetes ноды, конечно подключим Auto Scaling, но для начальной конфигурации всегда будем брать минимум 3 ноды. Возьмем инстансы m5.large, цены у каждого региона разные, но допустим, что нам нужно в самом дешевом – North Virginia. С учетом EBS при помощи расчета [AWS Pricing Calculator](#) получается 466 \$ per month.

Допустим, что мы будем использовать AWS RDS Postgres и DynamoDB для наших баз и так же с помощью [AWS Pricing Calculator](#) рассчитаем. Для RDS получится 3 инстанса db.m5.xlarge (по каждому для каждого окружения) – 510 \$ per month. Для DynamoDB приблизительно выделим 200 \$ per month.

Так же для удаленного хранения бекапов нам нужен AWS S3, приблизительно вбедим для него 200 \$ per month.

Terraform Cloud – 20 \$ per user / month. Представим, что наша инфраструктурная команда состоит из 3 человек, получаем 60 \$ per month.

Version Control System – 4 \$ per user / month. Цена одинакова для Gitlab и Github, если использовать не Enterprise решения, а обычное с небольшими наворотами) Допустим, что у нас в команде 15 человек и получаем 60 \$ per month.

Допустим, что нам так же нужны какие-то подписки на прочие ресурсы/инструменты/технологии и т.д. и выделим для этого 200 \$ per month.

В сумме для разворачивания начальной инфраструктуры нам необходимо ~2000 \$ в месяц.