Что нужно сделать перед тем, как преступить к заданию:

1) Установите IDE. Лучше всего будет VS Code

2) Установите гит, разберитесь как будете с ним работать. Git используем через терминал

3) Создайте репу (публичную) в гитхабе в которой вы будете выполнять свои практические задания (название - Фамилия\_analytics\_practice). Ссылку пришлете мне.

4) Создаем новую ветку для выполнения задания (task\_1)

5) После того как выполнили задание, коммитим и пушим. В main мерджить не надо

Задания:

1) Вы работаете в крупной компании которая продает строительные материалы и товары для дома (как прототип можно рассматривать Леруа). Компании реализует товары как в своих оффлайн гипермаркетах так и через интернет магазин. У компании есть собственные склады, а также есть склады поставщиков. У компании есть свои своя служба доставки, а также доставка через партнеров. В данный момент управление всей деятельностью компании (в том числе и управление всем бэкофисом) осуществляется с помощью монолитного приложения. Ваша задача распилить монолит на микросервисы. Что нужно сделать:

1) Составьте документ в формате AsciiDoc: - Опишите в документе подход который вы выберете для распиливания монолита - Опишите доменные области бизнеса - Опишите какие микросервисы вы выделите;

2) С помощью PlantUML составьте диаграммы С4 для вашей микросервисной архитектуры: - Контекстная диаграмма - Диаграмма контейнеров;

1.1) Ваши диаграммы интегрируйте в AsciiDoc документ.

- Экспортируете в формате SVG, и вставляете в свой документ изображение.

- Одну из бд на диаграмме контейнеров изобразите в виде кластера (с шардами и репликами)

- Добавьте кеширование в вашу диаграмму, где необходимо

- Одну из баз кэширования визуализируйте в виде кластера (Redis Cluster)

2) Начнем с проектирования БД для ваших микросервисов. Можете выбрать любые из ваших архитектурных диаграмм.:

- Мерджим task\_1 в main

- Делаем новую ветку в вашем git репозитории — task\_2

- Смотрим на диаграмму IE в PlantUML и готовим две диаграммы:

1. ER диаграмма для реляционной БД. Для этой задачи можем закрыть глаза на то, что у нас database per service. Просто готовим диаграмму со всеми отношениями, как будто у нас единая БД. Должно быть не менее 5 сущностей в диаграмме.

2. ER диаграмма для нереляционной БД. Описываем сущности как минимум двух коллекций.

3) Подключались к MongoDB Compass и там создавали документы и коллекции, использовали командную строку и встроенный функционал для этого;

4) Сделайте новую ветку в git. И в доке опишите ваши топики, подумайте как вы их назовете, что в них будет литься. Опишите сообщения в 3 форматах JSON, Avro, Protobuf;