**Вопросы к экзамену по курсу**

**«Микросервисная архитектура»**

**Тема: REST API - балансировка/распределение нагрузки между микросервисами**

1. Структура HTTP пакета:
   * методы запроса
   * request URI + query string
   * заголовки пакета
   * body - «тело» запроса

Описать особенности передачи данных в различных частях пакета. Какие из них могут использоваться балансировщиком для дальнейшей маршрутизации.

1. Протокол REST
   * команды (методы) прокола
   * соответствие команд REST методам HTTP протокола
   * передача параметров для выполнения запроса
   * коды ответов HTTP сервера (ошибки, успешное выполнение и др.)
   * результаты выполнения запроса, форматы JSON, Base64 и др.

Описать особенности проектирования сервисов в соответствии с протоколом REST. Выделить плюсы и минусы использования протокола REST.

1. Балансировка / распределение нагрузки
   * использование Nginx в качестве балансировщика (proxy)
   * маршрутизация с использованием location (эндпоинты)
   * условная маршрутизация IF
   * таблицы сопоставления MAP
   * группировка сервисов UPSTRAM
   * коды ответов HTTP сервера (ошибки, перенаправление и др.)

Описать основные принципы логической маршрутизации на основе сравнения заголовков (методов) запроса. Описать принципы распределения нагрузки с использованием Upstream (round-robin, weight и т.д.).

**Тема: Синхронные и асинхронные протоколы передачи данных**

1. Сравнительные характеристики синхронных / асинхронных протоколов
   * установка и поддержание соединения (timeout, keep-alive, ping/pong)
   * передача данных в заголовках либо в теле пакета
   * восстановление соединения (reconnect)

Описать основные принципы работы синхронных/асинхронных протоколов. Сравнить особенности передачи пакетов данных через разные типы соединений. Описать процессы установки/поддержания соединения и данных на примере протокола Websocket.

1. Протокол RPC (remote procedure call)
   * принципы проектирования RPC протокола
   * реализация RPC протокола поверх HTTP запросов
   * передача данных для выполнения RPC запроса
   * особенности маршрутизации/балансировки RPC запросов
   * передача результатов выполнения запроса

Описать особенности реализации RPC протокола (сравнительная характеристика с REST API). Передача параметров запроса / результатов ответа, форматы сериализации данных (на примере JSON).

1. Модель обмена сообщениями Publish / Subscribe (публикация / подписка)
   * принципы работы брокеров сообщений
   * публикация (Publish) сообщений
   * подписка (Subscribe) на темы (очереди) сообщений
   * модель realtime подписки / доставки
   * модель durable подписки / доставки

Описать основные принципы обмена сообщениями между сервисами по модели pub/sub. Особенности использования realtime брокеров сообщений. Особенности использования durable очередей (с возможностью длительного хранения и восстановления истории).

**Тема: Использование инструментов автоматической генерации кода для проектирования и разработки микросервисных приложений**

1. Использование протокола gRPC и формата Protobuf
   * стандарт protobuf для описания структур и функций протокола
   * описание структур в формате protobuf (типы данных, порядок полей и т.д.)
   * описание функций в формате protobuf
   * генерация кода модулей структур данных
   * генерация кода модулей клиентских и серверных приложений

Описать основные принципы обмена данными между микросервисами с использованием технологии gRPC. Описать особенности процесса проектирования сервисов и использования инструментов автоматической кодогенерации на основе protobuf.

1. Использование инструментов миграции для управления структурой базы данных
   * механизмы миграции реляционных БД
   * формат хранения файлов миграции
   * использование утилит миграции
   * проектирование архитектуры БД и организация командной работы с использованием механизмов миграции

Описать механизмы работы технологии миграции. Использование утилит миграции, версионирование БД. Какие аспекты, связанные с изменением структуры БД нужно учитывать при применении/откате миграций.

1. Использование инструментов автогенерации кода для работы с Базами данных
   * генерация кода модулей на основе миграций и SQL-запросов
   * использование утилит генерации кода (на примере SQLC)
   * проектирование архитектуры модулей приложения и организация командной работы с использованием кодогенерации

Описать механизмы работы утилит автоматической генерации кода модулей для реализации функций запросов к БД. Какие аспекты использования автогенерации мы должны учитывать при проектировании архитектуры сервисов и организации командной работы.

**Тема: Аутентификация в микросервисных приложениях. Технологии JWT и OpenId**

1. Проектирование системы аутентификации в соответствии со стандартом OpenId

Общие принципы

1. Использование JWT-токенов в процессах аутентификации

Общие принципы

1. Интеграция и федерация сервисов и систем с использованием технологий OpenId

Общие принципы

**Тема: Использование инструментов мониторинга микросервисных приложений. Технологии OpenTelemetry. Инструменты пакета Grafana.**

1. Инструменты профилирования и трассировки приложений
   * процесс профилирования, использование сервиса Pyroscope
   * процесс трассировки, использование сервиса Tempo
   * использование пакета Grafana для визуального представления данных
   * организация командной работы с инструментами профилирования и трассировки

Описать общие принципы мониторинга работы приложений. Процессы профилирования и трассировки, инструменты для сбора информации и визуального отображения результатов. Организация командной работы с использованием этих инструментов.

1. Использование метрик для мониторинга работы приложения
   * классификация метрик приложения (технические, эксплуатационные, бизнес-метрики и т.д.)
   * типы метрик
   * процессы сбора, агрегации и хранения метрик, пакет Prometheus
   * использование пакета Grafana для визуального представления данных
   * анализ и планирование на основе полученных метрик

Описать общие принципы мониторинга работы приложений. Процессы сбора метрик и анализ работы приложения. Использование системы Alert’ов для оповещения и управления работой компонентов приложения. Организация командной работы с использованием этих инструментов.

1. Использование системы логирования для сбора и анализа информации о работе приложения
   * процесс логирования событий в приложениии
   * уровни логирования (Info, Warn, Error, Debug)
   * сбор и отправка логов в систему хранения, пакет Promtail
   * хранение и обработка логов, пакет Loki
   * использование пакета Grafana для визуального представления данных
   * анализ логов

Описать общие принципы логирования событий в приложении. Процессы сбора, хранения и анализа информации о логировании. Использование системы Alert’ов для оповещения и управления работой компонентов приложения. Организация командной работы с использованием этих инструментов.