

# **Вопросы к экзамену**

## **Протоколы и маршрутизация**

### **1. Структура HTTP пакета**

Методы запроса, request URI + query string, заголовки пакета, body - «тело» запроса.

### **2. Протокол REST**

Команды (методы) протокола, 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.

### **3. Балансировка / распределение нагрузки**

Использование Nginx в качестве балансировщика (proxy), маршрутизация с использованием location (эндпоинты), условная маршрутизация IF, таблицы сопоставления MAP, группировка сервисов UPSTREAM, коды ответов HTTP сервера (ошибки, перенаправление и др.)

---

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

Передача данных в различных частях пакета. Какие из них могут использоваться балансировщиком для дальнейшей маршрутизации.

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

## **Мониторинг и профилирование работы приложения**

### **4. 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, анализ и планирование на основе полученных метрик.

### **5. Использование системы логирования для сбора и анализа информации о работе приложения**

Процесс логирования событий в приложении, уровни логирования (Info, Warn, Error, Debug), сбор и отправка логов в систему хранения, пакет Promtail, хранение и обработка логов. Анализ логов.

---

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

## **Системы виртуализации для запуска и управления приложениями**

### **6. Docker и виртуализация в контейнерах**

Что такое Docker, контейнеры, image? Разница между виртуализацией и контейнеризацией. Dockerfile, сборка приложений из исходного кода. Передача параметров через переменные окружения.

### **7. Kubernetes – система управления контейнерами**

Что такое Kubernetes? Stateless и statefull приложения. Основные компоненты системы Kubernetes. Основные сущности: Namespace, Pod, Deployment, ReplicaSet, DaemonSet, ConfigMap, Service, Ingress и т.д.

---

Описать основные принципы контейнерной виртуализации. Этапы сборки контейнеров из исходного кода приложений (разница между интерпретируемыми и компилируемыми языками). Публикация контейнеров в реестре.

Запуск и управление контейнерами в системе Kubernetes.

Привести примеры Dockerfile и манифестов Kubernetes с комментариями.