1. **«Мотивация».** Напишите здесь, почему в систему нужно добавить мониторинг и что это даст компании.

* Мониторинг позволит получить полное представление о работе системы на основе отслуживаемых метрик.
* С помощью инструментов наблюдения, команда быстрее сможет определить корень возникших проблем в системе.
* На основе собранных показателей, появится возможность выявления узких мест, что способствует принятия стратегических решений и формированию инсайдов для бизнеса.
* Мониторинг способствует своевременному выявлению и решению проблем, что улучшает пользовательский опыт.

1. **«Выбор подхода к мониторингу».** Выберите, какой подход к мониторингу вы будете использовать: RED, USE или «Четыре золотых сигнала». Для разных частей системы можно использовать разные подходы.

Система Shop, Система MES и Система CRM –подход «Четыре золотых сигнала». Мы предоставляем API для большого потока клиентов, данный подход позволяет заметить аномалии в работе заранее и быстро среагировать на ситуацию.

Messages Queue – подход «RED», даст нам представление о пропускной способности сервиса. Мы будем понимать, как часто нам поступают заказы, есть ли ошибки и узнаем количество времени на обработку запроса на взятие заказа в работу.

1. **Опишите, какие метрики и в каких частях системы вы будете отслеживать.**
2. Number of dead-letter-exchange letters in RabbitMQ - количество сообщений, попавших в Dead Letter Queue (DLQ), потому что они не были обработаны

Критический показатель: быстрое и значительное увеличение

Что делать:

• Проанализировать причину: неверный формат сообщений, истёкший TTL или переполненная очередь

• Увеличить TTL, пересмотреть логику обработки или оптимизировать Consumer

1. Number of message in flight in RabbitMQ - количество сообщений, которые RabbitMQ отправил, но не получил подтверждение от Consumer

Критический показатель: увеличение при нормальной работе Consumer указывает на проблемы.

Что делать:

• Проверить производительность Consumer

• Увеличить число Consumers или оптимизировать их обработку

1. Number of requests (RPS) for internet shop API
2. Number of requests (RPS) for CRM API
3. Number of requests (RPS) for MES API

количество запросов в секунду к API

Критический показатель: зависит от пропускной способности API, резкий рост может указывать на DDoS

Что делать:

• Включить rate-limiting

• Масштабировать API (добавить инстансы)

1. Number of requests (RPS) per user for internet shop API
2. Number of requests (RPS) per user for CRM API
3. Number of requests (RPS) per user for MES API

запросы в секунду, отправляемые одним пользователем

Критический показатель: высокая активность одного пользователя может быть подозрительной

Что делать:

• Настроить rate-limiting для пользователей

1. CPU % for shop API
2. CPU % for CRM API
3. CPU % for MES API

загруженность процессора инстансов API

Критический показатель: более 80–90% длительное время

Что делать:

• Добавить больше CPU или горизонтально масштабировать инстансы

• Оптимизировать код

1. Memory Utilisation for shop API
2. Memory Utilisation for CRM API
3. Memory Utilisation for MES API

использование оперативной памяти API

Критический показатель: постоянное превышение 80–90%

Что делать:

• Увеличить доступную память

• Оптимизировать использование памяти (например, снизить объём кеша)

1. Memory Utilisation for shop db instance
2. Memory Utilisation for MES db instance

потребление оперативной памяти базой данных

Критический показатель: 80–90%

Что делать:

• Проверить настройки кеширования и оптимизировать запросы

• Увеличить объём памяти

1. Number of connections for shop db instance
2. Number of connections for MES db instance

количество активных подключений к базе данных

Критический показатель: близость к максимальному лимиту подключения (например, 1000 для PostgreSQL по умолчанию)

Что делать:

• Настроить пул соединений (например, pgBouncer)

• Оптимизировать приложение, чтобы уменьшить количество соединений

1. Response time (latency) for shop API
2. Response time (latency) for CRM API
3. Response time (latency) for MES API

среднее время ответа API

Критический показатель: зависит от SLA, например, >100 мс

Что делать:

• Оптимизировать обработку запросов

• Масштабировать API или внедрить кэширование

1. Size of S3 storage

объём данных, хранящихся в S3

Критический показатель: увеличение стоимости из-за превышения лимитов хранения

Что делать:

• Оптимизировать хранение данных (архивирование, удаление старых файлов)

1. Size of shop db instance
2. Size of MES db instance

размер хранилища, используемого базой данных

Критический показатель: 80–90% доступного объёма

Что делать:

• Увеличить объём хранилища

• Архивировать или удалять старые данные

1. Number of HTTP 500 for shop API
2. Number of HTTP 500 for CRM API
3. Number of HTTP 500 for MES API

количество ошибок на стороне сервера

Критический показатель: увеличение

Что делать:

• Логировать и исправить причину ошибок

1. **«План действий».** Напишите высокоуровнево, какие задачи вы видите для реализации. Это будет драфт технического задания. Например, «Создать инстанс time-series базы с использованием такой-то технологии».
   1. Реализовать паттерн Backpressure, позволит формировать новые заказы в соответствии со скоростью их обработки через MES API.
   2. Настроить мониторинг и посмотреть показатели насыщенности , принять соответствующие меры см.п.3.
   3. Реализовать кеширование для MES, SHOP, CRM
   4. Настроить логирование и трейсинг
2. **Дополнительное задание.** Описала в п.3.