Опять же рассматриваю решения внедрения легирования, как часть общего решения «Наблюдаемости».

Т.е. у нас есть мониторинг и трейсинг.

Поэтому кол-во логов может быть сокращено без риска потерь функциональности, при этом будет сохранено место БД системы логирования.

1. **Проанализируйте систему компании и C4-диаграмму в контексте планирования логирования.**

Собираем логи из систем Интернет магазин, CRM,MES.

Логи делим на бизнес (заказ), пользовательские (действия пользователя), системные ошибки выполнения автоматизированных процессов внутри системы.

Набор бизнесовых и пользовательских логов будет частично пересекаться (по смыслу), но у них разные задачи.

Не важно в чем мы будем обрабатывать и хранить логи, сотрудник сервисной поддержки должен иметь возможность ответить на три вопроса.

Что по факту с заказом (например) статус.

Что делал пользователь (для примера опять же с этим заказом).

Что произошло не так. Сбой системы. Например оператор выбрал заказ, перевёл на себя, а система по какой-то причине не изменила статус заказа и не проставила исполнителя. Заказ застрял.

**Составьте список необходимых логов с уровнем INFO.**

INFO — это штатное поведение.

**Заказ (бизнесовая часть) система. (Пользователь логин из сертификата – избегаем любого упоминания перс данных)**

Заказ создан (указываем в логе систему из которой пришел заказ, пользователь системы)

Заказ изменен (изменение заказа оператором CRM)

Заказ передан (CRM)

Заказ получен (MES)

Заказ выбран оператором (MES)

Заказ в работе (если я правильно понимаю из всех систем)

Заказ выполнен (MES)

Заказ отправлен (MES)

Заказ закрыт (CRM)

**Пользовательские (пользователь логин из сертификата – избегаем любого упоминания перс данных) действия:**

Вход в систему (если нет sso через keyckloack(на диаграмме нет раздела), тогда логи в другой системе)

Если есть большое кол-во интерфейсов, переходы.

Запуск отчетов.

Создание, изменение, удаление объектов (например заказов).

**Системные**

В системе есть автоматизированные процессы, процессы делятся на стадии со своими статусами.

Вот эти стадии процессов мы и логируем.

Для примера процесс выбора оператором заказа. Примерно – проставление исполнителя – изменения статуса в работе – установка предварительной даты исполнения и т.п. Все эти стадии должны попасть в логи.

**Напишите, будете ли вы использовать другие уровни логирования и при каких обстоятельствах**

FATAL — обязательно. (+все API)

ERROR — обязательно, особенно в разделе системных логов (+все API)

WARN — не обязательно.

1. **Добавьте в документ раздел «Мотивация».**

Логирование это первое, с чего бы я начинал вообще весь процесс анализа и внедрение концепции «Наблюдаемости». Ее почему то всегда любят начинать с «Мониторинга», мое личное мнение, потому что с логированием, обычно вопрос более менее решен.

В нашем случае, если бы было реализовано логирование и отправка логов в формате JSON в Elasticsearch, тогда бы мы смогли разобраться в ситуациях с заказами проследив заказ и обрыв процесса через логи систем.

Логирование позволило бы понять, что происходило, когда и при каких обстоятельствах. Без логов сложно понять, из-за чего появляется периодическая (непостоянная) и только при определенных условиях ошибка.

**Команда не сможет реализовать единовременно логирование и трейсинг всех выделенных для этого систем. Поэтому опишите, для каких систем нужно настраивать логирование и трейсинг в первую очередь и почему.**

Я и не рассматриваю покрытие трейсингом и логированием всего и сразу. Выше в шапке пояснения.

Если совсем все плохо. Минимальный набор.

Интернет магазин оставляем в покое, реализация после обработки более важных систем. Оставляем на штатных логах. Это веб приложение и postgresql. В случае чего разберут в ручную.

Трейсинг заказа от CRM (номер и уникальные ид скорее всего присваиваются в CRM, сможем на них опираться) до MES (Попадание в список заказов MES считаем конечной точкой процесса. Сам MES фактически не трогаем, собираем ответ на отправку, что заказ попал в очередь MES. Если ответ не пришел или заказ не попал в систему считаем процесс в ошибке.)

Логирование JSON в Elasticsearch и разбор в Kibana делаем только для MES (само собой включая API), как для самой загруженной системы. Логи БД не трогаем. Состояние БД мониторится. В случае ошибок разбираем логи postgres.

Но это не лучший вариант.

1. **Добавьте раздел «Предлагаемое решение».**

Реализация через в распределенное хранилище на базе ELK (Elasticsearch, Logstash и Kibana).

Механизм сбора логов выглядит так:

Logstash собирает логи и переносит их в хранилище.

Elasticsearch помогает найти нужные строки в этих логах (разбор нашего JSON)

Kibana визуализирует.

Все три компонента разработаны на основе открытого кода, благодаря чему их можно модифицировать если потребуется.

**Проработайте политику безопасности в отношении логов**

Только сотрудники сервисной службы и админы и сотрудники ИБ могут иметь доступ к ELK.

Реализация через X-Pack, конфигурация AD в elasticsearch.yml (ssl.verification\_mode). Не очень красиво, но должно работать

Я бы рекомендовал ELK (Kibana) Active Directory/LDAP Authentication / keycloak (sso). Но на схеме не представлена аутентификация / авторизация.

Понятно, что можно использовать внутренние учетки созданные руками, но я, понятия не имею, как это обосновать ИБ. Там одних регламентов потребуется штук пять и процесс согласования новых пользователей для заведения.

**Проработайте политику хранения в отношении логов**

Отдельный индекс нужен, но на будущее когда все системы будут обложены логированием.

По хранению точного ответа дать не могу. Сначала надо посмотреть политики компании по персональным данным. Есть ли они в системах, можно ли признать эти данные справочными и т.п..

Возьмем минимальный срок хранения логов по перс данным - не менее трех лет.

Если в MES нет перс данных, а у нас логирование только по MES. До появления логов CRM и интернет магазина можно временно выставить минимальный срок хранения от 6-ти месяцев кажется.

**Проработайте необходимые мероприятия для превращения системы сбора логов в систему анализа логов:**

Предположительно использовать Elasticsearch и Kibana для сетевого мониторинга и оповещения (для API). Анализ сетевого трафика и журналов в реальном времени позволит быстро отреагировать на угрозу.

А за остальным пусть на первых парах следит мониторинг.